CSP-J 初赛模拟卷（第 01 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

1．中国的国家顶级域名是（ ）。

A．.cn　　B．.ch　　C．.chn　　D．.china

2．下列不属于面向对象程序设计语言的是（ ）。

A．C　　B．C++　　C．Java　　D．C#

3．以比较作为基本运算，在N个数中找出最大数，最坏情况下所需要的最少的比较次数为（ ）。

A．N^2　　B．N　　C．N-1　　D．N+1

4．表达式a*(b+c)*d 的后缀表达式为（ ），其中“*”和“+”是运算符。

A．**a+bcd　　B．abc+*d*　　C．abc+d**　　D．*a*+bcd

5．在 8 位二进制补码中，10101011 表示的数是十进制下的（ ）。

A．43　　B．-85　　C．-43　　D．-84

6．FTP 可以用于（ ）。

A．远程传输文件　　B．发送电子邮件　　C．浏览网页　　D．网上聊天

7．319和377的最大公约数是（ ）。

A．27　　B．33　　C．29　　D．31

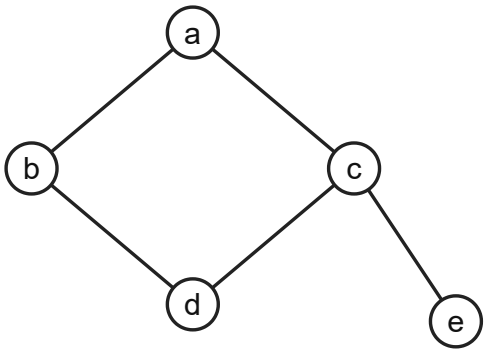
8．对于有n个顶点、m条边的无向联通图（m>n），需要删掉（ ）条边才能使其成为一棵树。

A．n-1　　B．m - n　　C．m - n - 1　　D．m - n + 1

9．如果一棵二叉树的中序遍历是 BAC，那么它的先序遍历不可能是（ ）。

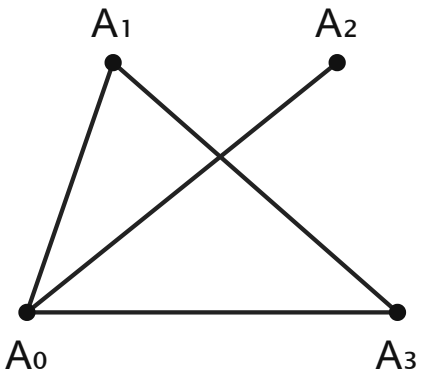
A．ABC　　B．CBA　　C．ACB　　D．BAC

10. 以a为起点，对右边的无向图进行深度优先遍历，则b,c,d,e四个点中有可能作为最后一个遍历到的点个数为（ ）。



A. 1　　B. 2　　C. 3　　D. 4

11. 以 A_0作为起点，对下面的无向图进行深度优先遍历时，遍历顺序不可能是（ ）。



A. A_0, A_1, A_2, A_3　　B. A_0, A_1, A_3, A_2　　C. A_0, A_2, A_1, A_3　　D. A_0, A_3, A_1, A_2

12. 6个人，两个人组一队，总共组成三只不区分队伍的编号，不同的组队情况有（ ）种。

A. 10　　B. 15　　C. 30　　D. 20

13. 如果根的高度为 1,具有 61 个结点的完全二叉树的高度为（ ）。

A. 5　　B. 6　　C. 7　　D. 8

14. 设x=true,y=true,z=false以下逻辑运算表达式值为真的是（ ）。

A. $(y \vee z) \wedge x \wedge z$　　B. $x \wedge (z \vee y) \wedge z$　　C. $(x \wedge y) \wedge z$　　D. $(x \wedge y) \vee (z \vee x)$

15. 有五副不同颜色的手套（共 10 只手套，每副手套左右手各 1 只），一次性从中取 6 只手套，请问恰好能配成两副手套的不同取法有（ ）种。

A. 120　　B. 180　　C. 150　　D. 30

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 √，错误填 ×；共 18 小题，共计 36 分，各小题分值见题目）

(1)

```
01 #include <cstdio>
02 #include <cstring>
03 using namespace std;
04 char st[100001];
05 int main() {
06     scanf("%s", st);
07     int n = strlen(st);
08     for (int i = 1; i <= n; ++i) {
09         if (n % i == 0) {
10             char c = st[i - 1];
11             if (c >= 'a')
12                 st[i - 1] = c - 'a' + 'A';
13         }
14     }
15     printf("%s", st);
16     return 0;
17 }
```

判断题

16. (1.5分) 输入的字符串只能由小写字母或大写字母组成。（ ）

17. (1.5分) 若将第 8 行的 i = 1 改为 i = 0，程序运行时会发生错误。（ ）

18. (1.5分) 若将第 8 行的 i <= n 改为 i * i <= n，程序运行结果不会改变。（ ）

19. (1.5分) 若输入的字符串全部由大写字母组成，那么输出的字符串就跟输入的字符串一样。（ ）

单选题

20. (3分) 若输入的字符串长度为 18，那么输入的字符串跟输出的字符串相比，至多有（ ）个字符不同。

A. 18　　B. 6　　C. 10　　D. 1

21. (3分) 若输入的字符串长度为（ ），那么输入的字符串跟输出的字符串相比，至多有 36 个字符不同。

A. 36　　B. 100000　　C. 1　　D. 128

(2)

```
#include <iostream>
using namespace std;

long long n, ans;
int k, len;
long long d[1000000];

int main() {
    cin >> n >> k;
    d[0] = 0;
    len= 1;
    ans = 0;
    for (long long i = 0; i <n; ++i) {
        ++d[0];
        for (int j = 0; j + 1<len; ++j) {
            if (d[j] == k) {
                d[j] = 0;
                d[j + 1] += 1;
                ++ans;
            }
        }
        if (d[len - 1] == k) {
            d[len - 1] = 0;
            d[len] =1;
            ++len;
            ++ans;
        }
```

```
    }
    cout << ans << endl;
    return 0;
}
```

假设输入的 n 是不超过 2^{62} 的正整数，k 是不超过 10000 的正整数，完成下面的判断题和单选题。

判断题

22. (1.5分) 若 $k=1$，则输出 ans 时，len = n。（ ）

23. (1.5分) 若 $k>1$，则输出 ans 时，len 一定小于 n。（ ）

24. (1.5分) 若 $k>1$，则输出 ans 时，k^{len} 一定大于 n。（ ）

单选题

25. (3分) 若输入的 $n=10^{15}$，$k=1$，则输出等于（ ）。

A. 1　　B. $(10^{30}-10^{15})/2$　　C. $(10^{30}+10^{15})/2$　　D. 10^{15}

26. (3分) 若输入的 $n=205891132094649$（即 3^{30}），$k=3$，则输出等于（ ）。

A. 3^{30}　　B. $(3^{30}-1)/2$　　C. $3^{30}-1$　　D. $(3^{30}+1)/2$

27. (3分) 若输入的 $n=100010002000090$，$k=10$，则输出等于（ ）。

A. 11112222444543　　B. 11122222444453

C. 11122222444543　　D. 11112222444453

(3)

```
#include <stdio.h>

int n;
int a[1000];

int f(int x)
{
    int ret = 0;
    for (; x; x &= x - 1) ret++;
    return ret;
}

int g(int x)
```

```
{
    return x & -x;
}

int main()
{
    scanf("%d", &n);
    for (int i = 0; i < n; i++) scanf("%d", &a[i]);
    for (int i = 0; i < n; i++)
        printf("%d ", f(a[i]) + g(a[i]));
    printf("\n");
    return 0;
}
```

判断题

28. (1.5分) 输入的n等于1001时，程序不会发生下标越界。（ ）

29. (1.5分) 输入的 $a[i]$ 必须全为正整数，否则程序将陷入死循环。（ ）

30. (1.5分) 当输入为“5 2 11 9 16 10”时，输出为“3 4 3 17 5”。（ ）

31. (1.5分) 当输入为”1 511998”时，输出为”18”。（ ）

32. (1.5分) 将源代码中 g 函数的定义(13-16行)移到main函数的后面，程序可以正常编译运行。（ ）

单选题

33. (3分) 当输入为”2 -65536 2147483647”时，输出为（ ）

A. “65532 33”　　B. “65552 32”　　C. “65535 34”　　D. “65554 33”

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **10** 小题，共计 **30** 分，各小题分值见题目）

(1)

（最小区间覆盖）给出 n 个区间，第 i 个区间的左右端点是 $[a_i, b_i]$。现在要在这些区间中选出若干个，使得区间 $[0, m]$ 被所选区间的并覆盖（即每一个 $0 \leq i < m$ 都在某个所选的区间中）。保证答案存在，求所选区间个数的最小值。

输入第一行包含两个整数 n 和 m （$1 \leq n \leq 5000, 1 \leq m \leq 10^9$）接下来 n 行，每行两个整数 a_i, b_i （$0 \leq a_i, b_i \leq m$）。

提示:采用贪心法解决这个问题。采用 $O(n^2)$ 的时间复杂度排序，然后贪心选择这些区间。试补全程序。

```
#include <iostream>

using namespace std;

const int MAXN = 5000;
int n, m;
struct segment { int a, b; } A[MAXN];

void sort() // 排序
{
    for (int i = 0; i < n; i++)
        for (int j = 1; j < n; j++)
            if (__(1)__)
            {
                segment t = A[j];
                __(2)__
            }
}

int main()
{
    cin >> n >> m;
    for (int i = 0; i < n; i++)
        cin >> A[i].a >> A[i].b;
    sort();
    int p = 1;
    for (int i = 1; i < n; i++)
        if (__(3)__)
            A[p++] = A[i];
    n = p;
    int ans =0, r = 0;
    int q = 0;
    while (r < m)
    {
        while (__(4)__)
            q++;
        __(5)__;
        ans++;
```

```
    }
    cout << ans << endl;
    return 0;
}
```

34. (3分) (1)处应填

A. A[j].b>A[j-1].b B. A[j].a<A[j-1].a
C. A[j].a>A[j-1].a D. A[j].b<A[j-1].b

35. (3分) (2)处应填

A. A[j+1]=A[j];A[j]=t; B. A[j-1]=A[j];A[j]=t;
C. A[j]=A[j+1];A[j+1]=t; D. A[j]=A[j-1];A[j-1]=t;

36. (3分) (3)处应填

A. A[i].b>A[p-1].b B. A[i].b<A[i-1].b
C. A[i].b>A[i-1].b D. A[i].b<A[p-1].b

37. (3分) (4)处应填

A. q+1<n&&A[q+1].a<=r B. q+1<n&&A[q+1].b<=r
C. q<n&&A[q].a<=r D. q<n&&A[q].b<=r

38. (3分) (5)处应填

A. r=max(r,A[q+1].b) B. r=max(r,A[q].b)
C. r=max(r,A[q+1].a) D. q++

(2)

（计数排序）计数排序是一个广泛使用的排序方法。下面的程序使用双关键字计数排序，将 n 对10000以内的整数，从小到大排序。例如有三对整数(3,4)、(2,4)、(3,3)那么排序之后应该是(2,4)、(3,3)、(3,4)。输入第一行为 n ，接下来 n 行，第 i 行有两个数 $a[i]$ 和 $b[i]$，分别表示第 i 对整数的第一关键字和第二关键字。从小到大排序后输出。
数据范围 $1 < n < 10^7, 1 < a[i], b[i] < 10^4$ 。提示:应先对第二关键字排序，再对第一关键字

排序。数组 ord[] 存储第二关键字排序的结果，数组 res[] 存储双关键字排序的结果。试补全程序。

```
01 #include <cstdio>
02 #include <cstring>
03 using namespace std;
04 const int maxn = 10000000;
05 const int maxs = 10000;
06
07 int n;
08 unsigned a[maxn], b[maxn],res[maxn], ord[maxn];
09 unsigned cnt[maxs + 1];
10 int main() {
11     scanf("%d", &n);
12     for (int i = 0; i < n; ++i)
13         scanf("%u%u", &a[i], &b[i]);
14     memset(cnt, 0, sizeof(cnt));
15     for (int i = 0; i < n; ++i)
16         __(1)__; // 利用 cnt 数组统计数量
17     for (int i = 0; i < maxs; ++i)
18         cnt[i + 1] += cnt[i];
19     for (int i = 0; i < n; ++i)
20         __(2)__; // 记录初步排序结果
21     memset(cnt, 0, sizeof(cnt));
22     for (int i = 0; i < n; ++i)
23         __(3)__; // 利用 cnt 数组统计数量
24     for (int i = 0; i < maxs; ++i)
25         cnt[i + 1] += cnt[i];
26     for (int i = n - 1; i >= 0; --i)
27         __(4)__; // 记录最终排序结果
28     for (int i = 0; i < n; i++)
29         printf("%u %u\n", __(5)__);
30
31     return 0;
32 }
```

39. (3分) (1)处应填

A. ++cnt [i]　　B. ++cnt[b[i]]

C. ++cnt[a[i] * maxs + b[i]]　　D. ++cnt[a[i]]

40. (3分) (2)处应填

A. ord[--cnt[a[i]]] = i　　B. ord[--cnt[b[i]]] = a[i]
C. ord[--cnt[a[i]]] = b[i]　D. ord[--cnt[b[i]]] = i

41. (3分) (3)处应填

A. ++cnt[b[i]]　B. ++cnt[a[i] * maxs + b[i]]
C. ++cnt[a[i]]　D. ++cnt [i]

42. (3分) (4)处应填

A. res[--cnt[a[ord[i]]]] = ord[i]　B. res[--cnt[b[ord[i]]]] = ord[i]
C. res[--cnt[b[i]]] = ord[i]　　D. res[--cnt[a[i]]] = ord[i]

43. (3分) (5)处应填

A. a[i], b[i]　　B. a[res[i]], b[res[i]]
C. a[ord[res[i]], b[ord[res[i]]　D. a[res[ord[i]], b[res[ord[i]])

CSP-J 初赛模拟卷（第 02 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

1．以下哪一种设备属于输出设备：（ ）

A．扫描仪　　B．键盘　　C．鼠标　　D．打印机

2．如果256 种颜色用二进制编码来表示，至少需要（ ）位。

A．6　　B．7　　C．8　　D．9

3．对于入栈顺序为a,b,c,d,e的序列，下列（ ）不是合法的出栈序列。

A．a,b,c,d,e　　B．e,d,c,b,a　　C．b,a,c,d,e　　D．c,d,a,e,b

4．如果一棵二叉树只有根结点，那么这棵二叉树高度为1。请问高度为5的完全二叉树有（ ）种不同形态?

A．16　　B．15　　C．17　　D．32

5．二进制数 11.01 在十进制下是（ ）。

A．3.25　　B．4.125　　C．6.25　　D．11.125

6．下列各无符号十进制整数中，能用八位二进制表示的数中最大的是（ ）。

A．296　　B．133　　C．256　　D．199

7．一些数字可以颠倒过来看，例如0、1、8颠倒过来还是本身，6颠倒过来是9，9颠倒过来看还是6,其他数字颠倒过来都不构成数字。 类似的，一些多位数也可以颠倒过来看，比如106颠倒过来是901。假设某个城市的车牌只由5位数字组成，每一位都可以取0到9。 请问这个城市最多有多少个车牌倒过来恰好还是原来的车牌？（ ）

A．60　　B．125　　C．75　　D．100

8．二进制数101.11对应的十进制数是（ ）。

A．6.5　　B．5.5　　C．5.75　　D．5.25

9．使用冒泡排序对序列进行升序排列，每执行一次交换操作系统将会减少 1 个逆序对，因此序列 5，4，3，2，1 需要执行（ ）次操作，才能完成冒泡排序。

A．0　　B．5　　C．10　　D．15

10. 由1, 1, 2, 2, 3这五个数字组成不同的三位数有（ ）种。

A. 18　　B. 15　　C. 12　　D. 24

11. 将（2,6,10,17）分别存储到某个地址区间为0~10的哈希表中，如果哈希函数 $h(x)=$（ ），将不会产生冲突，其中a mod b表示a除以b的余数。

A. x mod 11　　B. $x^2 \mod 11$

C. (2x) mod 11　　D. $\lfloor sqrt(x) \rfloor \mod 11$，其中 $\lfloor sqrt(x) \rfloor$ 表示 $sqrt(x)$ 下取整

12. 寄存器是（ ）的重要组成部分.

A. 硬盘　　B. 高速缓存　　C. 内存　　D. 中央处理器（CPU）

13. 某算法的时间复杂度满足递推式 $T(n)=T(n-1)+n$，边界条件为 $T(0)=1$，则该算法的时间复杂度为（ ）。

A. $O(\log n)$　　B. $O(n\log n)$　　C. $O(n)$　　D. $O(n^2)$

14. 100以内最大的素数是（ ）。

A. 89　　B. 97　　C. 91　　D. 93

15. 言泡排序算法的伪代码如下:

```
输入：数组L, n ≥ k。输出：按非递减顺序排序的 L。
算法 BubbleSort:
    1. FLAG ← n //标记被交换的最后元素位置
    2. while FLAG > 1 do
    3. k ← FLAG -1
    4. FLAG ← 1
    5. for j=1 to k do
    6. if L(j) > L(j+1) then do
    7. L(j) ⇌ L(j+1)
    8. FLAG ← j
```

对 n 个数用以上言泡排序算法进行排序，最少需要比较多少次？（ ）。

A. n^2　　B. $n-2$　　C. $n-1$　　D. n

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 **√**，错误填 **×**；共 **18** 小题，共计 **40.5** 分，各小题分值见题目）

(1)

```
01 #include <stdio.h>
02 #include <string.h>
03
04 char base[64];
05 char table[256];
06 char str[256];
07 char ans[256];
08
09 void init()
10 {
11     for (int i = 0; i < 26; i++) base[i] = 'A' + i;
12     for (int i = 0; i < 26; i++) base[26 + i] = 'a' + i;
13     for (int i = 0; i < 10; i++) base[52 + i] = '0' + i;
14     base[62] = '+', base[63] = '/';
15
16     for (int i = 0; i < 256; i++) table[i] = 0xff;
17     for (int i = 0; i < 64; i++) table[base[i]] = i;
18     table['='] = 0;
19 }
20
21 void decode(char *str)
22 {
23     char *ret = ans;
24     int i, len = strlen(str);
25     for (i = 0; i < len; i += 4) {
26         (*ret++) = table[str[i]] << 2 | table[str[i + 1]] >> 4;
27         if (str[i + 2] != '=')
28             (*ret++) = (table[str[i + 1]] & 0x0f) << 4 | table[str[i
    + 2]] >> 2;
29         if(str[i + 3] != '=')
30             (*ret++) = table[str[i + 2]] << 6 | table[str[i + 3]];
31     }
32 }
33
34 int main()
35 {
36     init();
37     printf("%d\n", (int)table[0]);
```

```
38
39      scanf("%s", str);
40      decode(str);
41      printf("%s\n", ans);
42      return 0;
43  }
```

假设 char 是 8 位有符号类型，输入字符串长度为 4 的倍数，且输入不超过数组范围。

判断题

16. (1.5分) 输出的第二行一定是由小写字母、大写字母、数字和 +、/、= 构成的字符串。（ ）

17. (1.5分) 可能存在输入不同，但输出的第二行相同的情形。（ ）

18. (1.5分) 输出的第一行为 -1。（ ）

单选题

19. (3分) 设输入字符串长度为 n，decode 函数的时间复杂度为（ ）。

A. $O(\sqrt{n})$　　B. $O(n)$　　C. $O(n\log n)$　　D. $O(n^2)$

20. (3分) 当输入为 Y3Nx 时，输出的第二行为（ ）。

A. csp　　B. csq　　C. CSP　　D. Csp

21. (3.5分) 当输入为 Y2NmIDIwMjE= 时，输出的第二行为（ ）。

A. ccf2021　　B. ccf2022　　C. ccf 2021　　D. ccf 2022

(2)

```
01  #include <algorithm>
02  #include <iostream>
03  using namespace std;
04
05  int n;
06  int d[50][2];
07  int ans;
08
09  void dfs(int n, int sum) {
10      if (n == 1) {
11          ans = max(sum, ans);
12          return;
```

```
13      }
14      for (int i = 1; i < n; ++i) {
15          int a = d[i - 1][0], b = d[i - 1][1];
16          int x = d[i][0], y = d[i][1];
17          d[i - 1][0] = a + x;
18          d[i - 1][1] = b + y;
19          for (int j = i; j < n - 1; ++j)
20              d[j][0] = d[j + 1][0], d[j][1] = d[j + 1][1];
21          int s = a + x + abs(b - y);
22          dfs(n - 1, sum + s);
23          for (int j = n - 1; j > i; --j)
24              d[j][0] = d[j - 1][0], d[j][1] = d[j - 1][1];
25          d[i - 1][0] = a, d[i - 1][1] = b;
26          d[i][0] = x, d[i][1] = y;
27      }
28  }
29
30  int main() {
31      cin >> n;
32      for (int i = 0; i < n; ++i)
33          cin >> d[i][0];
34      for (int i = 0; i < n;++i)
35          cin >> d[i][1];
36      ans = 0;
37      dfs(n, 0);
38      cout << ans << endl;
39      return 0;
40  }
```

假设输入的 n 是不超过50的正整数，$d[i][0]$、$d[i][i]$ 都是不超过10000的正整数，完成下面的判断题和单选题：

判断题

22. (1.5分) 若输入 n 为0，此程序可能会死循环或发生运行错误。（ ）

23. (1.5分) 若输入 n 为20,接下来的输入全为0，则输出为0。（ ）

24. (1.5分) 输出的数一定不小于输入的 $d[i][0]$ 和 $d[i][1]$ 的任意一个。（ ）

单选题

25. (3分) 若输入的 n 为 20，接下来的输入是 20 个 9 和 20 个 0，则输出为（ ）。

A. 1890　　B. 1881　　C. 1908　　D. 1917

26. (3分) 若输入的 n 为30,接下来的输入是30个0和30个5,则输出为（ ）。

A. 2000　　B. 2010　　C. 2030　　D. 2020

27. (4分) 若输入的 n 为 15，接下来的输入是 15 到 1，以及 15 到 1，则输出为（ ）。

A. 2440　　B. 2220　　C. 2240　　D. 2420

(3)

```
01 #include<cstdio>
02 using namespace std;
03 int n, m;
04 int a[100], b[100];
05
06 int main() {
07     scanf("%d%d", &n, &m);
08     for (int i = 1; i <= n; ++i)
09         a[i] = b[i] = 0;
10     for (int i = 1; i <= m; ++i) {
11         int x, y;
12         scanf("%d%d", &x, &y);
13         if (a[x] < y && b[y] < x) {
14             if (a[x] > 0)
15                 b[a[x]] = 0;
16             if (b[y] > 0)
17                 a[b[y]] = 0;
18             a[x] = y;
19             b[y] = x;
20         }
21     }
22     int ans = 0;
23     for (int i = 1; i <= n; ++i) {
24         if (a[i] == 0)
25             ++ans;
26         if (b[i] == 0)
27             ++ans;
```

```
    }
    printf("%d", ans);
    return 0;
}
```

假设输入的 n 和 m 都是正整数，x 和 y 都是在 $[1,n]$ 的范围内的整数，完成下面的判断题和单选题：

判断题

28. (1.5分) 当 $m>0$ 时，输出的值一定小于 $2n$。（ ）

29. (1.5分) 执行完第27行的“+ + ans”时，ans一定是偶数。（ ）

30. (1.5分) $a[i]$ 和 $b[i]$ 不可能同时大于0。（ ）

31. (1.5分) 右程序执行到第13行时， x 总是小于 y ，那么第15行不会被执行。（ ）

单选题

32. (3分) 若 m 个 x 两两不同，且 m 个 y 两两不同，则输出的值为（ ）

A. $2n-2m$　　B. $2n+2$　　C. $2n-2$　　D. $2n$

33. (3分) 若 m 个 x 两两不同，且 m 个 y 都相等，则输出的值为（ ）

A. $2n-2$　　B. $2n$　　C. $2m$　　D. $2n-2m$

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **10** 小题，共计 **30** 分，各小题分值见题目）

(1)

（矩阵变幻）有一个奇幻的矩阵，在不停的变幻，其变幻方式为:数字0变成矩阵 0 0 0 1

数字1变成矩阵 1 1 1 0

最初该矩阵只有一个元素 0，变幻 n 次后，矩阵会变成什么样？

例如，矩阵最初为:[0]； 矩阵变幻 1 次后: 0 0 0 1

矩阵变幻 2 次后: 0 0 0 0 0 1 0 1 0 0 1 1 0 1 1 0

输入一行一个不超过10的正整数n。输出变幻n次后的矩阵。

试补全程序。

提示：

<< 表示二进制左移运算符，例如 $(11)_2 << 2 = (1100)_2$ ；而^表示二进制异或运算符，它将两个参与运算的数中的每个对应的二进制位进行比较，若两个二进制位相同，则运算结果的对应二进制位为0，反之为1。

```
#include <cstdio>
using namespace std;
int n;
const int max_size = 1 << 10;

int res[max_size][max_size];

void recursive(int x, int y, int n, int t) {
    if (n == 0) {
        res[x][y] = __(1)__;
        return;
    }
    int step = 1 << (n - 1);
    recursive(__(2)__, n - 1, t);
    recursive(x, y + step, n - 1, t);
    recursive(x + step, y, n - 1, t);
    recursive(__(3)__, n - 1, !t);
}

int main() {
    scanf("%d", &n);
    recursive(0, 0, __(4)__);
    int size = __(5)__;
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++)
            printf("%d", res[i][j]);
        puts("");
    }
    return 0;
}
```

34. (3分) (1)处应填

A. n%2　　B. 0　　C. t　　D. 1

35. (3分) (2)处应填

A. x-step,y-step　B. x,y-step　C. x-step,y　D. x,y

36. (3分) (3)处应填

A. x-step,y-step　B. x+step,y+step　C. x-step,y　D. x,y-step

37. (3分) (4)处应填

A. n-1,n%2　　B. n, 0　　C. n,n%2　　D. n-1,0

38. (3分) (5)处应填

A. $1 \ll (n+1)$　　B. 1<<n　　C. $n+1$　　D. 1<<(n-1)

(2)

（矩形计数）平面上有*n*个关键点，求有多少个四条边都和*x*轴或者*y*轴平行的矩形，满足四个顶点都是关键点。给出的关键点可能有重复，但完全重合的矩形只计一次。 试补全枚举算法。

```
01 #include <stdio.h>
02
03 struct point {
04     int x, y, id;
05 };
06
07 int equals(struct point a, struct point b){
08     return a.x == b.x && a.y == b.y;
09 }
10
11 int cmp(struct point a, struct point b){
12     return  __(1)__;
13 }
14
15 void sort(struct point A[], int n){
16     for (int i = 0; i < n; i++)
17         for (int j = 1; j < n; j++)
18             if (cmp(A[j], A[j - 1])){
19                 struct point t = A[j];
20                 A[j] = A[j - 1];
21                 A[j - 1] = t;
22             }
23 }
24
25 int unique(struct point A[], int n){
26     int t = 0;
27     for (int i = 0; i < n; i++)
```

```
28          if (__(2)__)
29              A[t++] = A[i];
30      return t;
31  }
32
33  int binary_search(struct point A[], int n, int x, int y){
34      struct point p;
35      p.x = x;
36      p.y = y;
37      p.id = n;
38      int a = 0, b = n - 1;
39      while(a < b){
40          int mid = __(3)__;
41          if (__(4)__)
42              a = mid + 1;
43          else
44              b = mid;
45      }
46      return equals(A[a], p);
47  }
48
49  #define MAXN 1000
50  struct point A[MAXN];
51
52  int main(){
53      int n;
54      scanf("%d", &n);
55      for (int i = 0; i < n; i++){
56          scanf("%d %d", &A[i].x, &A[i].y);
57          A[i].id = i;
58      }
59      sort(A, n);
60      n = unique(A, n);
61      int ans = 0;
62      for (int i = 0; i < n; i++)
63          for (int j = 0; j < n; j++)
64              if ( __(5)__ && binary_search(A, n, A[i].x, A[j].y) &&
    binary_search(A, n, A[j].x, A[i].y)){
65                  ans++;
66              }
67      printf("%d\n", ans);
68      return 0;
69  }
```

39. (3分) (1)处应填

A. a.x != b.x ? a.x < b.x : a.id < b.id
B. a.x != b.x ? a.x < b.x : a.y < b.y
C. equals(a,b) ? a.id < b.id : a.x < b.x
D. equals(a,b) ? a.id < b.id : (a.x != b.x ? a.x < b.x : a.y < b.y)

40. (3分) (2)处应填

A. i == 0 || cmp(A[i], A[i-1])　　B. t == 0 || equals(A[i], A[t-1])
C. i == 0 || !cmp(A[i], A[i-1])　　D. t == 0 || !equals(A[i], A[t-1])

41. (3分) (3)处应填

A. b - (b-a)/2 + 1　　B. (a+b+1) >> 1
C. (a+b) >> 1　　D. a + (b-a+1)/2

42. (3分) (4)处应填

A. !cmp(A[mid], p)　　B. cmp(A[mid], p)
C. cmp(p, A[mid])　　D. !cmp(p, A[mid])

43. (3分) (5)处应填

A. A[i].x == A[j].x
B. A[i].id < A[j].id
C. A[i].x == A[j].x && A[i].id < A[j].id
D. A[i].x < A[j].x && A[i].y < A[j].y

CSP-J 初赛模拟卷（第 03 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

1．以下逻辑表达式的值恒为真的是（　）。

A．P ∨ (¬P ∧ Q) ∨ (¬P ∧ ¬Q)
B．Q ∨ (¬P ∧ Q) ∨ (P ∧ ¬Q)
C．P ∨ Q ∨ (P ∧ ¬Q) ∨ (¬P ∧ Q)
D．P ∨ ¬Q ∨ (P ∧ ¬Q) ∨ (¬P ∧ ¬Q)

2．在 8 位二进制补码中，10101011 表示的数是十进制下的（　）。

A．43　　B．-85　　C．-43　　D．-84

3．二进制数 11.01 在十进制下是（　）。

A．3.25　　B．4.125　　C．6.25　　D．11.125

4．把 64 位非零浮点数强制转换成32 位浮点数后，不可能（　）。

A．大于原数　B．小于原数　C．等于原数　D．与原数符号相反

5．提出“存储程序”的计算机工作原理的是（　）。

A．克劳德·香农　　B．戈登·摩尔　　C．查尔斯·巴贝奇　　D．冯·诺依曼

6．如果开始时计算机处于小写输入状态，现在有一只小老鼠反复按照 CapsLock、 字母键 A、字母键 S 和字母键 D 的顺序循环按键，即 CapsLock、A、S、D、CapsLock、A、S、D……，屏幕上输出的第 81 个 字符是字母（　）。

A．A　　B．S　　C．D　　D．a

7．原字符串中任意一段连续的字符所组成的新字符串称为子串。 则字符“AAABBBCCC”共有（　）个不同的非空子串。

A．3　　B．12　　C．36　　D．45

8．广度优先搜索时，需要用到的数据结构是（　）。

A．链表　　B．队列　　C．栈　　D．散列表

9．通常在搜索引擎中，对某个关键词加上双引号表示（　）。

A．排除关键词，不显示任何包含该关键词的结果
B．将关键词分解，在搜索结果中必须包含其中的一部分
C．精确搜索，只显示包含整个关键词的结果
D．站内搜索，只显示关键词所指向网站的内容

10．将（2，6，10，17）分别存储到某个地址区间为0~10 的哈希表中，如果哈希函数=（　），将不会产生冲突，其中表示a除以b的余数。

A．x mod 11　　B．x^2 mod 11
C．$(2x)$ mod 11　　D．$\lfloor sqrt(x) \rfloor$ mod 11，其中 $\lfloor sqrt(x) \rfloor$ 表示sqrt(x)下取整

11．考虑如下递归算法，则调用solve(7)得到的返回结果为（　）。

```
01 solve(n)
02 if n<=1 return 1
03 else if n>=5 return n*solve(n-2)
04 else return n*solve(n-1)
```

A．105　　B．840　　C．210　　D．420

12．给定一个含N个不相同数字的数组，在最坏情况下，找出其中最大或最小的数，至少需要 N-1 次比较操作。则最坏情况下，在该数组中同时找最大与 最小的数至少需要（　）次比较操作。（⌈ ⌉表示向上取整，⌊ ⌋表示向下取整）

A．$\lceil 3N/2 \rceil - 2$　　B．$\lfloor 3N/2 \rfloor - 2$　　C．$2N - 2$　　D．$2N - 4$

13．一家四口人，至少两个人生日属于同一月份的概率是（　）（假定每个人生日属于每个月份的概率相同且不同人之间相互独立）。

A．1/12　　B．1/144　　C．41/96　　D．3/4

14．一棵二叉树的前序遍历序列是ABCDEFG，后序遍历序列是CBFEGDA，则根结点的左子树的结点个数可能是（　）。

A．2　　B．3　　C．4　　D．5

15．2017 年 10 月 1 日是星期日，1999 年 10 月 1 日是（　）。

A．星期三　　B．星期日　　C．星期五　　D．星期二

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 **√**，错误填 **×**；共 **18** 小题，共计 **38.5** 分，各小题分值见题目）

(1)

```
01 #include <cstdlib>
02 #include <iostream>
03 using namespace std;
04
05 char encoder[26] = {'C','S','P',0};
06 char decoder[26];
07
08 string st;
09
10 int main()  {
11     int k = 0;
12     for (int i = 0; i < 26; ++i)
13         if (encoder[i] != 0) ++k;
14     for (char x ='A'; x <= 'Z'; ++x) {
15         bool flag = true;
16         for (int i = 0; i < 26; ++i)
17             if (encoder[i] ==x) {
18                 flag = false;
19                 break;
20             }
21         if (flag) {
22             encoder[k]= x;
23             ++k;
24         }
25     }
26     for (int i = 0; i < 26; ++i)
27         decoder[encoder[i]- 'A'] = i + 'A';
28     cin >> st;
29     for (int i = 0; i < st.length( ); ++i)
30         st[i] = decoder[st[i] -'A'];
31     cout << st;
32     return 0;
33 }
```

判断题

16. (1.5分) 输入的字符串应当只由大写字母组成，否则在访问数组时可能越界。（　）

17. (1.5分) 若输入的字符串不是空串，则输入的字符串与输出的字符串一定不一样。（ ）

18. (1.5分) 将第 12 行的 i < 26 改为 i < 16，程序运行结果不会改变。（ ）

19. (1.5分) 将第 26 行的 i < 26 改为 i < 16，程序运行结果不会改变。（ ）

单选题

20. (3分) 若输出的字符串为 ABCABCABCA，则下列说法正确的是（ ）。

A. 输入的字符串中既有 S 又有 P　B. 输入的字符串中既有 S 又有 B
C. 输入的字符串中既有 A 又有 P　D. 输入的字符串中既有 A 又有 B

21. (3分) 若输出的字符串为 CSPCSPCSPCSP，则下列说法正确的是（ ）。

A. 输入的字符串中既有 P 又有 A　B. 输入的字符串中既有 S 又有 R
C. 输入的字符串中既有 A 又有 S　D. 输入的字符串中既无 S 又无 A

(2)

```
01 #include<cstdio>
02 #include<cstring>
03 using namespace std;
04 int n, m;
05 int f[101][101];
06 int F[101][101];
07 int main(){
08     scanf("%d%d",&n,&m);//n的值在1到100之间
09     memset(f,-1,sizeof(f));
10     for(int i = 1; i <=  m; i++) {
11         int u,v,w;//w的值在0到10000之间
12         scanf("%d%d%d",&u,&v,&w);
13         f[u][v]=f[v][u]=w;
14     }
15     for(int k = 1; k <= n; k++)
16         for(int i=1;i <=n; i++)
17             for(int j=1;j<= n; j++)
18                 if(f[i][k]!=-1&&f[k][j]!=-1)
19                     if(f[i][j]==-1||f[i][j]>f[k][j]+f[i][k])
20                         f[i][j]=f[i][k]+f[k][j];
21     int ans=2147483647;
22     for(int i=1;i<=n;i++)
23         for(int j=1;j<=n; j++){
24             for(int x=1;x<=n; x++)
```

```
25                  for(int y=1;y<=n; y++)
26                      F[x][y]=f[x][y];
27              F[i][j]=F[j][i]=0;
28              for(int x=1;x<= n; x++)
29                  for(int y=1;y<=n; y++)
30                      if(F[x][y]==-1||F[x][y]>F[x][i]+F[i][y])
31                          F[x][y]=F[x][i]+F[i][y];
32              for(int x=1;x<=n; x++)
33                  for(int y=1;y<=n; y++)
34                      if(F[x][y]==-1||F[x][y]>F[x][j]+F[j][y])
35                          F[x][y]=F[x][j]+F[j][y];
36              int res=0;
37              for(int x=1;x<=n; x++)
38                  for(int y=1;y<x; y++)
39                      res +=F[x][y];
40              ans = res < ans ? res : ans;
41          }
42      printf("%d\n",ans);
43      return 0;
44  }
```

判断题

22. (1分) 14到16行，将外层到内层的循环变量依次调整为i、j、k，程序的运行的结果不变。（ ）

23. (2分) 这个程序的时间复杂度和m无关。（ ）

24. (2分) 20行的ans如果初始化为10^7时，可能无法得到正确结果。（ ）

25. (2分) 若将第27到30行的部分和31到34行的两个部分互换，程序的运行的结果不变。（ ）

单选题

26. (3分) 若输入数据为4 5/1 2 3/1 3 6/2 3 4/2 4 7/3 4 2(其中“/”为换行符)，则输出为（ ）。

A. 14　　B. 18　　C. 21　　D. 28

27. (3分) 这个程序使用了（ ）算法。

A. Floyd　　B. Dijkstra　　C. Prim　　D. Kruskal

(3)

```
01 #include <iostream>
02 using namespace std;
03 const int V = 101;
04 int n, m, ans, e[V][V];
05 bool visited[V];
06 void dfs(int x, int len)
07 {
08     int i;
09     visited[x] = true;
10     if (len > ans) ans = len;
11     for (i = 1; i <= n; i++)
12         if ((!visited[i]) && (e[x][i] != -1))
13             dfs(i, len + e[x][i]);
14     visited[x] = false;
15 }
16 int main()
17 {
18     int i, j, a, b, c;
19     cin >> n >> m;
20     for (i = 1; i <= n; i++)
21         for (j = 1; j <= n; j++)
22             e[i][j] = -1;
23     for (i = 1; i <= m; i++) {
24         cin >> a >> b >> c;
25         e[a][b] = c;
26         e[b][a] = c;
27     }
28     for (i = 1; i <= n; i++)
29         visited[i] = false;
30     ans = 0;
31     for (i = 1; i <= n; i++)
32         dfs(i, 0);
33     cout << ans << endl;
34     return 0;
35 }
```

输入表示一个没有重边、自环的无向图，顶点编号为 **1** 到 n，且 $1 \leq n \leq 100$、边权为正整数。

判断题

28. (1.5分) 第 19 行的输入中，如果满足 $m = n(n-1)/2$，则第 20～22 行的初始化可以省略。（ ）

29. (1.5分) 将第 31 行改为 `for (i = n; i >= 1; i--)`，程序结果不受影响。（ ）

单选题

30. (1.5分) 若输入如下，则程序输出为（ ）。

```
4 2
1 2 1
3 4 1
```

A. 8　　B. 4　　C. 2　　D. 1

31. (3分) 若输入如下，则程序输出为（ ）。

```
4 6
1 2 1
2 3 1
3 4 1
4 1 1
1 3 1
2 4 1
```

A. 1　　B. 3　　C. 5　　D. 7

32. (3分) 若输入如下，则程序输出为（ ）。

```
4 3
1 2 10
2 3 20
3 1 30
```

A. 30　　B. 50　　C. 90　　D. 120

33. (3分) 若输入如下，则程序输出为（　）。

```
4 6
1 2 10
2 3 20
3 4 30
4 1 40
1 3 50
2 4 60
```

A. 60　　B. 100　　C. 150　　D. 200

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **10** 小题，共计 **29** 分，各小题分值见题目）

(1)

(封禁xxs)现有n个xxs(编号为1到n)，每个xxs都有一个关注者，第i个xxs的关注者是a_i。现在管理员要将其中的一些xxs的账号封禁，但需要注意的是如果封禁了第i个人，那么为了不打草惊蛇，就不能封禁他的关注者a_i。现在想知道最多可以封禁多少个xxs。 输入第一行是一个不超过300000的整数n，第二行是n个1到n的整数表示。 输出一行，一个整数表示答案。

```
01 #include<cstdio>
02 using namespace std;
03 #define MAXN 300005
04 int n, ans=0, a[MAXN], in[MAXN]={0};
05 bool vis[MAXN]={0};
06 void dfs(int cur, int w){
07     if(vis[cur])
08         return;
09     vis[cur]=true;
10     if(w==1) ans++;
11     __(1)__
12     if(__(2)__)
13         dfs(a[cur], __(3)__);
14 }
15 int main(){
16     scanf("%d",&n);
17     for(int i=1;i <=n; i++) {
18         scanf("%d", &a[i]);
19         in[a[i]]++;
20     }
```

```
21      for(int i=1;i <= n; i++)
22          if(!in[i]) __(4)__;
23      for(int i=1;i<=n; i++)
24          if(__(5)__) dfs(i,0);
25      printf("%d\n",ans);
26      return 0;
27  }
```

34. (3分) (1)处应填

A. a[cur]=cur; B. in[a[cur]]=0; C. in[a[cur]]--; D. in[cur]--;

35. (3分) (2)处应填

A. in[a[cur]] != 0 || w == 1 B. in[a[cur]==0 | w==0
C. in[a[cur]!=0 | w==0 D. in[a[cur]]==0 | w==1

36. (3分) (3)处应填

A. 0 B. 1 C. w D. 1-w

37. (3分) (4)处应填

A. dfs(i,1) B. dfs(i, 0) C. dfs(a[i], 1) D. dfs(a[i], 0)

38. (3分) (5)处应填

A. !in[i] B. in[i] C. !vis[i] D. vis[l]

(2)

完善程序:

（二叉查找树）二叉查找树具有如下性质:每个节点的值都大于其左子树上所有节点的值。小于其右子树上所有节点的值。试判断一棵树是否为二叉查找树。输入的第一行包含一个整数 n，表示这棵树有 n 个顶点，编号分别为 $1,2,\ldots,n$，其中编号为1的为根结点。之后的第1行有三个数value, left_child, right_child，分别表示该节点关键字的值、左子节点的编号、右子节点的编号；如果不存在左子节点或右子节点，则用0代替。输出1表示这棵树是二叉查找树，输出0则表示不是。

```
01  #include <iostream>
02  using namespace std;
03  const int SIZE = 100;
04  const int INFINITE = 1000000;
05  struct node
06  {
```

```
07      int left_child, right_child, value;
08  }; node a[SIZE];
09  int is_bst( int root, int lower_bound, int upper_bound )
10  {
11      int cur;
12      if ( root == 0 )
13          return(1);
14      cur = a[root].value;
15      if ( (cur > lower_bound) && ( __(1)__ ) && (is_bst(
    a[root].left_child, lower_bound, cur ) == 1) && (is_bst( __(2)__,
    __(3)__, __(4)__ ) == 1) )
16          return(1);
17      return(0);
18  }
19  int main()
20  {
21      int i, n; cin >> n;
22      for ( i = 1; i <= n; i++ )
23          cin >> a[i].value >> a[i].left_child >> a[i].right_child;
24      cout << is_bst( __(5)__, -INFINITE, INFINITE ) << endl;
25      return(0);
26  }
```

P/Q	Q为true	Q为false
P为true	P=1&&Q=1	P=1&&Q=0
P为false	P=0&&Q=1	P=0&&Q=0

39. (3分) (1)处应填：_____

40. (3分) (2)处应填：_____

41. (3分) (3)处应填：_____

42. (3分) (4)处应填：_____

43. (2分) (5)处应填：_____

CSP-J 初赛模拟卷（第 04 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

1．Linux下可执行文件的默认扩展名为（　）。

A．exe　　B．com　　C．dll　　D．以上都不是

2．冒泡排序算法的伪代码如下。 输入：长度为 n 的数组 L。输出：按非递减顺序排序的 L。

```
01 FLAG ← n
02 while FLAG > 1 do
03     k ← FLAG - 1
04     FLAG ← 1
05     for j = 1 to k do
06         if L(j) > L(j+1) then
07             L(j) ↔ L(j+1)
08             FLAG ← j
```

对 n 个数用以上冒泡排序算法进行排序，最少需要比较多少次？（ ）

A．n^2　　B．n - 2　　C．n - 1　　D．n

3．地址总线的位数决定了 CPU 可直接寻址的内存空间大小，例如地址总线为 16 位，其最大的可寻址空间为 64KB。如果地址总线是 32 位，则理论上最大可寻址的内存空间为（　）。

A．128KB　　B．1MB　　C．1GB　　D．4GB

4．无向完全图是图中每对顶点之间都恰好有一条边的简单图。已知无向完全图G有7个顶点，则它共有（　）条边。

A．7　　B．21　　C．42　　D．49

5．下列说法正确的是（　）。

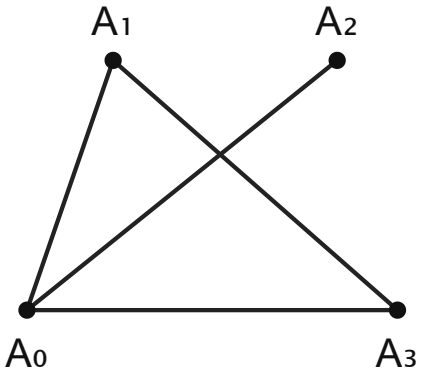
A．CPU的主要任务是执行数据运算和程序控制

B．存储器具有记忆能力，其中信息任何时候都不会丢失

C．两个显示器屏幕尺寸相同，则它们的分辨率必定相同

D．个人用户只能使用 Wifi 的方式连接到 Internet

6．以A_0作为起点，对下面的无向图进行深度优先遍历时，遍历顺序不可能是（　）。



A．A_0, A_1, A_2, A_3　　B．A_0, A_1, A_3, A_2　　C．A_0, A_2, A_1, A_3　　D．A_0, A_3, A_1, A_2

7．完全二叉树的顺序存储方案，是指将完全二叉树的结点从上至下、从左至右依次存放到一个顺序结构的数组中。假定根结点存放在数组的1号位置，则第k号结点的父结点如果存在的话，应当存放在数组的（　）号位置。

A．2k　　B．2k+1　　C．k/2下取整　　D．(k+1)/2下取整

8．下列几个32位IP地址中，书写错误的是（　）。

A．162.105.135.27　　B．192.168.0.1　　C．256.256.129.1　　D．10.0.0.1

9．设 X 、 Y 、 Z 分别代表三进制下的一位数字，若等式 $XY + ZX = XYX$ 在三进制下成立，那么同样在三进制下，等式$XY * ZX = ()$ 也成立。

A．YXZ　　B．ZXY　　C．XYZ　　D．XZY

10．有7个一模一样的苹果，放到3个一样的盘子中，盘子可以为空，一共有（　）种放法。

A．7　　B．8　　C．21　　D．3^7

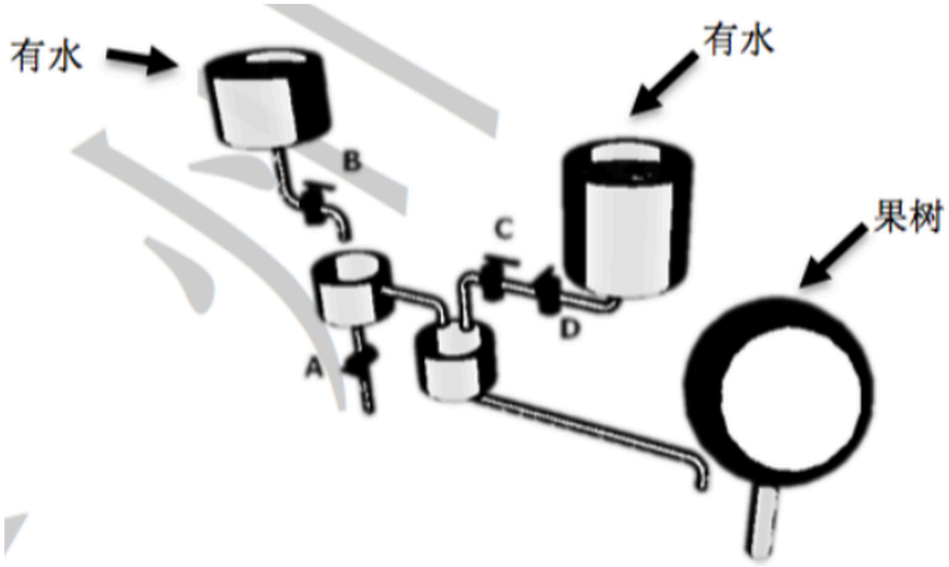
11．现有一段文言文，要通过二进制哈夫曼编码进行压缩。简单起见，假设这段文言文只由4个汉字“之”、“乎”、“者”、“也”组成，它们出现的次数分别为700、600、300、400。若待合并结点权值相同，优先选取由其他结点合并形成的结点。那么，“也”字的编码长度可能是（　）。

A．1　　B．2　　C．3　　D．4

12．某算法的计算时间表示为递推关系式 $T(n) = T(n-1) + n(n$ 为正整数$)$及 $T(0) = 1$，则该算法的时间复杂度为（　）。

A．$O(\log n)$　　B．$O(n \log n)$　　C．$O(n)$　　D．$O(n^2)$

13. 下图表示一个果园灌溉系统，有 A、B、C、D 四个阀门，每个阀门可以打开或关上，所有管道粗细相同，以下设置阀门的方法中，可以让果树浇上水的是（ ）。



A. B 打开，其他都关上　B. AB 都打开，CD 都关上
C. A 打开，其他都关上　D. D 打开，其他都关上

14. 10000 以内，与 10000 互质的正整数有（ ）个。

A. 2000　B. 4000　C. 6000　D. 8000

15. 一些数字可以颠倒过来看，例如0、1、8颠倒过来还是本身，6颠倒过来是9，9颠倒过来看还是6,其他数字颠倒过来都不构成数字。 类似的，一些多位数也可以颠倒过来看，比如106颠倒过来是901。假设某个城市的车牌只由5位数字组成，每一位都可以取0到9。 请问这个城市最多有多少个车牌倒过来恰好还是原来的车牌？（ ）

A. 60　B. 125　C. 75　D. 100

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 √，错误填 ×；共 18 小题，共计 42 分，各小题分值见题目）

(1)

```
01 #include<iostream>
02 using namespace std;
03 int main()
04 {
05     const int SIZE=10;
06     int height[SIZE],num[SIZE],n,ans;
07     cin>>n;
```

```
08      for(int i=0;i<n;i++){
09          cin>>height[i];
10          num[i]=1;
11          for(int j=0;j<i;j++){
12              if((height[j]<height[i])&&(num[j]>=num[i]))
13                  num[i]=num[j]+1;
14          }
15      }
16      ans=0;
17      for(int i=0;i<n;i++){
18          if(num[i]>ans) ans=num[i];
19      }
20      cout<<ans<<endl;
21      return 0;
22  }
```

判断题

16. (1.5分) 如果height数组中的输入有负数，则程序会出错。（ ）

17. (1.5分) 程序输出的 ans 小于或等于n。（ ）

18. (1.5分) 将12行num[j]>=num[i]“改为”num[j]>num[i]“，程序的输出结果不会改变。（ ）

19. (1.5分) 将18行”num[i]>ans”改为”num[i]>=ans”，程序的输出结果不会改变。（ ）

单选题

20. (3分) 若输入的数据为 10 1 1 1 1 1 1 1 1 1 则程序的输出结果是（ ）。

A. 4　　B. 3　　C. 2　　D. 1

21. (3分) 若输人的数据为 10 3 2 5 11 12 7 4 10 15 6 因程序的输出结果是（ ）。

A. 4　　B. 5　　C. 6　　D. 7

(2)

```
01 #include <iostream>
02 using namespace std;
03 const int maxn = 10000;
04 int n;
05 int a[maxn];
06 int b[maxn];
07 int f(int l, int r, int depth) {
08     if (l > r)
09         return 0;
10     int min = a[l], mink = l;
11     for (int i = l; i <= r; ++i) {
12         if (min > a[i]) {
13             min = a[i];
14             mink = i;
15         }
16     }
17     int lres = f(l, mink - 1, depth + 1);
18     int rres = f(mink + 1, r, depth + 1);
19     return lres + rres + depth * b[mink];
20 }
21 int main() {
22     cin >> n;
23     for (int i = 0; i < n; ++i)
24         cin >> a[i];
25     for (int i = 0; i < n; ++i)
26         cin >> b[i];
27     cout << f(0, n - 1, 1) << endl;
28     return 0;
29 }
```

判断题

22. (1.5分) 如果a数组有重复的数字，则程序运行时会发生错误。（ ）

23. (1.5分) 如果b数组全为0，则输出为0。（ ）

单选题

24. (3分) 当 $n = 100$ 时，最坏情况下，与第12行的比较运算执行的次数最接近的是:（ ）。

A. 5000　　B. 600　　C. 6　　D. 100

25. (3分) 当n=100时，最好情况下，与第12行的比较运算执行的次数最接近的是:（ ）。

A. 100　　B. 6　　C. 5000　　D. 600

26. (3分) 当 $n = 10$ 时，若 b 数组满足，对任意 $0 <= i < n$, 都有 $b[i] = i + 1$, 那么输出最大为（ ）。

A. 386　　B. 383　　C. 384　　D. 385

27. (4分) 当n=100时，若b数组满足，对任意 $0 <= i < n$ ，都有 $b[i] = 1$ ，那么输出最小为（ ）。

A. 582　　B. 580　　C. 579　　D. 581

(3)

```
01 #include<bits/stdc++.h>
02 using namespace std;
03 #define MOD 19260817
04 #define MAXN 1005
05 long long A[MAXN][MAXN]={0},sum[MAXN][MAXN]={0};
06 int n, m, q;
07 int main(){
08     A[1][1]=A[1][0]=1;
09     for(int i=2;i <= 1000; i++) {
10         A[i][0]=1;
11         for(int j = 1; j <= i; j++)
12             A[i][j]=(A[i-1][j]+A[i-1][j-1])%MOD;
13     }
14     for(int i =1;i <= 1000; i++)
15         for(int j=1;j<=1000;j++)
16             sum[i][j]=(sum[i-1][j]+sum[i][j-1]
17             -sum[i-1][j-1]+A[i][j]+MOD)%MOD;
18             int q;
19     cin >> q;
20     while(q--){
21         int n, m;
22         cin >> n >> m;
23         cout << sum[n][m] << endl;
24     }
25     return 0;
26 }
```

判断题

28. (1分) 当i<=j时，A[i][i]的值是0。（ ）

29. (2分) 当i>j时，A[i][j]的值相当于从i个不同元素中取出j个元素的排列数。（ ）

30. (2分) sum[i][j]的值(1<j<i≤1000)不小于sum[i-1][j-1]的值。（ ）

31. (2分) 若将第12行改为“A[i][j]=(A[i-1][j]+A[i-1][j-1]+MOD)%MOD;”，程序的运行的结果不变。（ ）

单选题

32. (4分) A[i][j] (1≤i≤10, 1≤j≤10)的所有元素中，最大值是（ ）。

A. 126　　B. 276　　C. 252　　D. 210

33. (3分) 若输入数据为 1/5 3(其中“/”为换行符)，则输出为（ ）。

A. 10　　B. 35　　C. 50　　D. 24

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **10** 小题，共计 **29** 分，各小题分值见题目）

(1)

（切割绳子）有 n条绳子，每条绳子的长度已知且均为正整数。绳子可以以任意正整数长度切割，但不可以连接。现在要从这些绳子中切割出 m 条长度相同的绳段，求绳段的最大长度是多少。 输入：第一行是一个不超过 100 的正整数 n，第二行是 n 个不超过10^6的正整数，表示每条绳子的长度，第三行是一个不超过10^8的正整数m。 输出：绳段的最大长度，若无法切割，输出Failed。

```
01 #include<iostream>
02 using namespace std;
03 int n, m, i, lbound, ubound, mid, count;
04 int len[100]; // 绳子长度
05 int main()
06 {
07     cin >> n;
08     count = 0;
09     for (i = 0; i < n; i++)
10     {
11         cin >> len[i];
```

```
12          __(1)__;
13      }
14      cin >> m;
15      if (__(2)__)
16      {
17          cout << "Failed" << endl;
18          return 0;
19      }
20      lbound = 1;
21      ubound = 1000000;
22      while (__(3)__)
23      {
24          mid = __(4)__;
25          count = 0;
26          for (i = 0; i < n; i++)
27              __(5)__;
28          if (count < m)
29              ubound = mid - 1;
30          else
31              lbound = mid;
32      }
33      cout << lbound << endl;
34      return 0;
35  }
```

34. (2.5分) (1)处应填：_____

35. (2.5分) (2)处应填：_____

36. (3分) (3)处应填：_____

37. (3分) (4)处应填：_____

38. (3分) (5)处应填：_____

(2)

完善程序

（排列数）输入两个正整数 n ， $m(1<n<20,1<m<n)$ ，在 $1\sim n$ 中任取 m 个数，按字典序从小到大输出所有这样的排列。

例如:输入:32

输出：

```
#include <iostream>
#include <cstring>
using namespace std;
const int SIZE =25;
bool used[SIZE];
int data[SIZE];
int n,m,i,j,k;
bool flag;
int main()
{
    cin>>n>>m;
    memset(used,false,sizeof(used));
    for(i=1;i<=m;i++)
    {
        data[i]=i;
        used[i]=true;
    }
    flag=true;
    while(flag)
    {
        for(i=1;i<=m-1;i++) cout<<data[i]<<" ";
        cout<<data[m]<<endl;
        flag= __(1)__ ;
        for(i=m;i>=1;i--)
        {
            __(2)__;
            for(j=data[i]+1;j<=n;j++)
                if(!used[j])
                {
                    used[j]=true;
                    data[i]=__(3)__ ;
                    flag=true;
                    break;
                }
            if(flag)
            {
                for(k=i+1;k<=m;k++)
                    for(j=1;j<= __(4)__;j++)
                        if(!used[j])
                        {
                            data[k]=j;
```

```
42                              used[j]=true;
43                              break;
44                          }
45                  __(5)__;
46              }
47          }
48      }
49      return 0;
50  }
```

39. (3分) (1)处应填：_____

40. (3分) (2)处应填：_____

41. (3分) (3)处应填：_____

42. (3分) (4)处应填：_____

43. (3分) (5)处应填：_____

CSP-J 初赛模拟卷（第 05 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

1．中国计算机学会于（　）年创办全国青少年计算机程序设计竞赛。

A．1983　　B．1984　　C．1985　　D．1986

2．一个 32 位整型变量占用（　）个字节。

A．4　　B．8　　C．32　　D．128

3．下列四个不同进制的数中，与其它三项数值上不相等的是

A．(269)16　B．(617)10　C．(1151)8　D．(1001101011)2

4．根节点深度为0，一棵深度为h的满 $k(k>1)$ 又树，即除最后一层无任何子节点外，每一层上的所有结点都有 k 个子结点的树，共有（　）个结点。

A．$(k^{h+1}-1)/(k-1)$　B．k^{h-1}　C．k^h　D．$k^{h-1}/(k-1)$

5．若有如下程序段，其中 s,a,b,c 均已定义为整型变量，且 a,c 均已赋值 （c大于0）。 `s = a; for (b = 1;b <= c; b++ ) s = s + 1;` 则与上述程序段修改 s 值的功能等价的赋值语句是（　）。

A．`s = a + b;`　　B．`s = a + c;`　　C．`s = s + c;`　　D．`s = b + c;`

6．现有一张分辨率为2048×1024 像素的 32 位真彩色图像。请问要存储这张图像，需要多大的存储空间？（　）。

A．16MB　　B．4MB　　C．8MB　　D．2MB

7．设简单无向图 G 有 16 条边且每个顶点的度数都是 2，则图 G 有（　）个顶点。

A．10　　B．12　　C．8　　D．16

8. 要求以下程序的功能是计算：s=1+1/2+1/3+…+1/10.

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    float s;
    s = 1.0;
    for(n = 10; n > 1; n--)
        s = s + 1 / n;
    cout << s << endl;
    return 0;
}
```

程序运行后输出结果错误,导致错误结果的程序行是（ ）。

A. s = 1.0;　　B. for(n = 10; n > 1; n--)

C. s = s + 1 / n;　　D. cout << s << endl;

9. 在 8 位二进制补码中，10101011 表示的数是十进制下的（ ）。

A. 43　　B. -85　　C. -43　　D. -84

10. 有五副不同颜色的手套（共 10 只手套，每副手套左右手各 1 只），一次性从中取 6 只手套，请问恰好能配成两副手套的不同取法有（ ）种。

A. 120　　B. 180　　C. 150　　D. 30

11. 如果根的高度为 1,具有 61 个结点的完全二叉树的高度为（ ）。

A. 5　　B. 6　　C. 7　　D. 8

12. 下列几个32位IP地址中，书写错误的是（ ）。

A. 162.105.135.27　　B. 192.168.0.1　　C. 256.256.129.1　　D. 10.0.0.1

13. 寄存器是（ ）的重要组成部分.

A. 硬盘　　B. 高速缓存　　C. 内存　　D. 中央处理器（CPU）

14. 下列协议中与电子邮件无关的是（ ）。

A. POP3　　B. SMTP　　C. WTO　　D. IMAP

15. 周末小明和爸爸妈妈三个人一起想动手做三道菜。小明负责洗菜、爸爸负责切菜、妈妈负责炒菜。假设做每道菜的顺序都是：先洗菜10分钟，然后切菜10分钟，最后炒菜10分钟。那么做一道菜需要30分钟。注意：两道不同的菜的相同步骤不可以同时进行。例如第一道菜和第二道的菜不能同时洗，也不能同时切。那么做完三道菜的最短时间需要（ ）分钟。

A. 90　　B. 60　　C. 50　　D. 40

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 **√**，错误填 **×**；共 **18** 小题，共计 **37.5** 分，各小题分值见题目）

(1)

```
01 #include <cstdlib>
02 #include <iostream>
03 using namespace std;
04
05 char encoder[26] = {'C','S','P',0};
06 char decoder[26];
07
08 string st;
09
10 int main()  {
11     int k = 0;
12     for (int i = 0; i < 26; ++i)
13         if (encoder[i] != 0) ++k;
14     for (char x ='A'; x <= 'Z'; ++x) {
15         bool flag = true;
16         for (int i = 0; i < 26; ++i)
17             if (encoder[i] ==x) {
18                 flag = false;
19                 break;
20             }
21         if (flag) {
22             encoder[k]= x;
23             ++k;
24         }
25     }
26     for (int i = 0; i < 26; ++i)
27         decoder[encoder[i]- 'A'] = i + 'A';
28     cin >> st;
29     for (int i = 0; i < st.length( ); ++i)
```

```
30          st[i] = decoder[st[i] -'A'];
31      cout << st;
32      return 0;
33  }
```

判断题

16. (1.5分) 输入的字符串应当只由大写字母组成，否则在访问数组时可能越界。（ ）

17. (1.5分) 若输入的字符串不是空串，则输入的字符串与输出的字符串一定不一样。（ ）

18. (1.5分) 将第 12 行的 `i < 26` 改为 `i < 16`，程序运行结果不会改变。（ ）

19. (1.5分) 将第 26 行的 `i < 26` 改为 `i < 16`，程序运行结果不会改变。（ ）

单选题

20. (3分) 若输出的字符串为 ABCABCABCA，则下列说法正确的是（ ）。

A. 输入的字符串中既有 S 又有 P　B. 输入的字符串中既有 S 又有 B
C. 输入的字符串中既有 A 又有 P　D. 输入的字符串中既有 A 又有 B

21. (3分) 若输出的字符串为 CSPCSPCSPCSP，则下列说法正确的是（ ）。

A. 输入的字符串中既有 P 又有 A　B. 输入的字符串中既有 S 又有 R
C. 输入的字符串中既有 A 又有 S　D. 输入的字符串中既无 S 又无 A

(2)

```
01  #include <cstdio>
02  using namespace std;
03  int n;
04  int a[100];
05
06  int main() {
07      scanf("%d", &n);
08      for (int i = 1; i <= n; ++i)
09          scanf("%d", &a[i]);
10      int ans = 1;
11      for (int i = 1; i <= n; ++i) {
12          if (i > 1 && a[i] < a[i - 1])
13              ans = i;
14          while (ans < n && a[i] >= a[ans + 1])
15              ++ans;
```

```
16          printf("%d\n", ans);
17      }
18      return 0;
19  }
```

判断题

22. (1分) 第16行输出ans时，ans的值一定大于i。（ ）

23. (1分) 程序输出的ans小于等于n。（ ）

24. (1.5分) 若将第12行的 $<$ 改为 $!=$ ，程序输出的结果不会改变。（ ）

25. (1.5分) 当程序执行到第16行时，若ans-i>2,则 $a[i+1] \leq a[i]$ 。（ ）

单选题

26. (3分) 若输入的a数组是一个严格单调递增的数列，此程序的时间复杂度（ ）

A. O(log n)　　B. O(n^2)　　C. O(n log n)　　D. O(n)

27. (4分) 最坏情况下，此程序的时间复杂度是（ ）

A. O(n^2)　　B. O(log n)　　C. O(n)　　D. O(n log n)

(3)

```
01  #include <iostream>
02  using namespace std;
03  const int V = 101;
04  int n, m, ans, e[V][V];
05  bool visited[V];
06  void dfs(int x, int len)
07  {
08      int i;
09      visited[x] = true;
10      if (len > ans) ans = len;
11      for (i = 1; i <= n; i++)
12          if ((!visited[i]) && (e[x][i] != -1))
13              dfs(i, len + e[x][i]);
14      visited[x] = false;
15  }
16  int main()
17  {
18      int i, j, a, b, c;
19      cin >> n >> m;
```

```
20      for (i = 1; i <= n; i++)
21          for (j = 1; j <= n; j++)
22              e[i][j] = -1;
23      for (i = 1; i <= m; i++) {
24          cin >> a >> b >> c;
25          e[a][b] = c;
26          e[b][a] = c;
27      }
28      for (i = 1; i <= n; i++)
29          visited[i] = false;
30      ans = 0;
31      for (i = 1; i <= n; i++)
32          dfs(i, 0);
33      cout << ans << endl;
34      return 0;
35  }
```

输入表示一个没有重边、自环的无向图，顶点编号为 1 到 n，且 $1 \leq n \leq 100$、边权为正整数。

判断题

28. (1.5分) 第 19 行的输入中，如果满足 $m = n(n-1)/2$，则第 20～22 行的初始化可以省略。（ ）

29. (1.5分) 将第 31 行改为 `for (i = n; i >= 1; i--)`，程序结果不受影响。（ ）

单选题

30. (1.5分) 若输入如下，则程序输出为（ ）。

```
4 2
1 2 1
3 4 1
```

A. 8　　B. 4　　C. 2　　D. 1

31. (3分) 若输入如下，则程序输出为（　）。

```
4 6
1 2 1
2 3 1
3 4 1
4 1 1
1 3 1
2 4 1
```

A. 1　　B. 3　　C. 5　　D. 7

32. (3分) 若输入如下，则程序输出为（　）。

```
4 3
1 2 10
2 3 20
3 1 30
```

A. 30　　B. 50　　C. 90　　D. 120

33. (3分) 若输入如下，则程序输出为（　）。

```
4 6
1 2 10
2 3 20
3 4 30
4 1 40
1 3 50
2 4 60
```

A. 60　　B. 100　　C. 150　　D. 200

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **10** 小题，共计 **31** 分，各小题分值见题目）

(1)

（最小区间覆盖）给出 n 个区间，第 i 个区间的左右端点是 $[a_i, b_i]$。现在要在这些区间中选出若干个，使得区间 $[0, m]$ 被所选区间的并覆盖（即每一个 $0 \leq i \leq m$ 部在某个所选的区间中）。保证答案存在，求所选区间个数的最小值。

输入第一行包含两个整数 n 和 m （$1 \leq n \leq 5000, 1 \leq m \leq 10^3$）接下来 n 行，每行两个整数 $a_i, b_i (0 \leq a_i, b_i \leq m)$。提示:使用贪心法解决这个问题。先用 $O(n^2)$ 的时间复杂度排

序，然后贪心选择这些区间。试补全程序。

```
#include <iostream>

using namespace std;

const int MAXN = 5000;
int n, m;
struct segment { int a, b; } A[MAXN];

void sort() // 排序
{
    for (int i = 0; i < n; i++)
        for (int j = 1; j < n; j++)
            if (__(1)__)
            {
                segment t = A[j];
                __(2)__
            }
}

int main()
{
    cin >> n >> m;
    for (int i = 0; i < n; i++)
        cin >> A[i].a >> A[i].b;
    sort();
    int p = 1;
    for (int i = 1; i < n; i++)
        if (__(3)__)
            A[p++] = A[i];
    n = p;
    int ans =0, r = 0;
    int q = 0;
    while (r < m)
    {
        while (__(4)__)
            q++;
        __(5)__;
        ans++;
```

```
    }
    cout << ans << endl;
    return 0;
}
```

34. (3分) (1)处应填

A. A[j].b>A[j-1].b B. A[j].a<A[j-1].a
C. A[j].a>A[j-1].a D. A[j].b<A[j-1].b

35. (3分) (2)处应填

A. A[j+1]=A[j];A[j]=t; B. A[j-1]=A[j];A[j]=t;
C. A[j]=A[j+1];A[j+1]=t; D. A[j]=A[j-1];A[j-1]=t;

36. (3分) (3)处应填

A. A[i].b>A[p-1].b B. A[i].b<A[i-1].b
C. A[i].b>A[i-1].b D. A[i].b<A[p-1].b

37. (3分) (4)处应填

A. q+1<n&&A[q+1].a<=r B. q+1<n&&A[q+1].b<=r
C. q<n&&A[q].a<=r D. q<n&&A[q].b<=r

38. (3分) (5)处应填

A. r=max(r,A[q+1].b) B. r=max(r,A[q].b)
C. r=max(r,A[q+1].a) D. q++

(2)

完善程序:

(最大子矩阵和)给出 m 行 n 列的整数矩阵，求最大的子矩阵和(子矩阵不能为空)。

输入第一行包含两个整数 m 和 n ，即矩阵的行数和列数。之后 m 行，每行 n 个整数，描述整个矩阵。程序最终输出最大的子矩阵和。比如在如下这个矩阵中:

```
4 4
0 -2 -7 0
9 2 -6 2
-4 1 -4 1
-1 8 0 -2
```

拥有最大和的子矩阵为：

```
9 2
-4 1
-1 8
```

其和为15

```
3 3
-2 10 20
-1 100 -2
0 -2 -3
```

最大子矩阵和为128

```
4 4
0 -2 -9 -9
-9 11 5 7
-4 -3 -7 -6
-1 7 7 5
```

最大子矩阵和为26

```
01 #include <iostream>
02 using namespace std;
03 const int SIZE = 100;
04 int matrix[SIZE + 1][SIZE + 1];
05 int rowsum[SIZE + 1][SIZE + 1]; /* rowsum[i][j]记录第i行前j个数的和
   */
06 int m, n, i, j, first, last, area, ans;
07 int main()
08 {
09     cin >> m >> n;
10     for ( i = 1; i <= m; i++ )
11         for ( j = 1; j <= n; j++ )
12             cin >> matrix[i][j];
13     ans = matrix   __(1)__;
14     for ( i = 1; i <= m; i++ )
15         __(2)__;
16     for ( i = 1; i <= m; i++ )
17         for ( j = 1; j <= n; j++ )
18             rowsum[i][j] = __(3)__;
19     for ( first = 1; first <= n; first++ )
20         for ( last = first; last <= n; last++ )
21         {
22             __(4)__;
```

```
23              for ( i = 1; i <= m; i++ )
24              {
25                  area += __(5)__;
26                  if ( area > ans )
27                      ans = area;
28                  if ( area < 0 )
29                      area = 0;
30              }
31          }
32      cout << ans << endl;
33      return(0);
34  }
```

39. (3分) (1)处应填：_____

40. (3分) (2)处应填：_____

41. (3分) (3)处应填：_____

42. (3分) (4)处应填：_____

43. (4分) (5)处应填：_____

CSP-J 初赛模拟卷（第 06 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

1．NOI 的中文意思是（ ）。

A．中国信息学联赛　　B．全国青少年信息学奥林匹克竞赛
C．中国青少年信息学奥林匹克竞赛　　D．中国计算机协会

2．1TB代表的字节数是（ ）。

A．2的10次方　　B．2的20次方　　C．2的30次方　　D．2的40次方

3．二进制数11 1011 1001 0111和01 0110 1110 1011进行逻辑与运算的结果是（ ）。

A．01 0010 1000 1011　　B．01 0010 1001 0011
C．01 0010 1000 0001　　D．01 0010 1000 0011

4．如果根结点的深度记为1，则一棵恰有2011个叶结点的二叉树的深度最少是（ ）。

A．10　　B．11　　C．12　　D．13

5．给定含有 n 个互不相同元素的单峰数组 L，满足先严格递增、后严格递减，峰顶位于数组内部。用下面的二分算法寻找峰顶，请将 a、b、c 三行代码依次填入空白处。正确的填空顺序是（ ）。

```
a. return Search(k + 1, r)
b. return Search(l, k - 1)
c. return L[k]

Search(l, r):
    if l == r then return L[l]
    k = floor((l + r) / 2)
    if L[k] > L[k - 1] and L[k] > L[k + 1]: ______
    else if L[k] < L[k + 1]: ______
    else: ______

调用 Search(2, n - 1)
```

A．c,a,b　　B．c,b,a　　C．a,b,c　　D．b,a,c

6．10 个三好学生名额分配到 7 个班级，每个班级至少有一个名额，一共有（ ）种不同的分配方案。

A．84　　B．72　　C．56　　D．504

7．6 个顶点的连通图的最小生成树，其边数为（ ）。

A．6　　B．5　　C．7　　D．4

8．设变量x为非负float型且已赋值，则以下语句中能将x中的数值保留到小数点后两位，并将第三位四舍五入的是（ ）。

A．x = (x * 100) + 0.5 / 100.0;　　B．x = (x * 100 + 0.5) / 100.0;
C．x = (int)(x * 100 + 0.5)/100.0;　　D．x = (x / 100 + 0.5) * 100.0;

9．二进制数 00100100 和 00010100 的和是（ ）。

A．00101000　　B．01100111　　C．01000100　　D．00111000

10．以下关于字符串的判定语句中正确的是（ ）。

A．字符串是一种特殊的线性表　　B．串的长度必须大于零
C．字符串不可以用数组来表示　　D．空格字符组成的串就是空串

11．（ ）是一种先进先出的线性表。

A．栈　　B．队列　　C．哈希表（散列表）　　D．二叉树

12．设 $x = true, y = true, z = false$，以下逻辑运算表达式值为真的是（ ）。

A．$(y \vee z) \wedge x \wedge z$　　B．$x \wedge (z \vee y) \wedge z$　　C．$(x \wedge y) \wedge z$　　D．$(x \wedge y) \vee (z \vee x)$

13．以下哪一种设备属于输出设备（ ）。

A．扫描仪　　B．键盘　　C．鼠标　　D．打印机

14．在 Windows 资源管理器中，用鼠标右键单击一个文件时，会出现一个名为“复制”的操作选项，它的意思是（ ）。

A．用剪切板中的文件替换该文件
B．在该文件所在文件夹中，将该文件克隆一份
C．将该文件复制到剪切板，并保留原文件
D．将该文件复制到剪切板，并删除原文件

15．甲、乙、丙三位同学选修课程，从 4 门课程中，甲选修 2 门，乙、丙各选修3门，则不同的选修方案共有（ ）种。

A．36　　B．48　　C．96　　D．192

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 **√**，错误填 **×**；共 **18** 小题，共计 **39.5** 分，各小题分值见题目）

(1)

```
01 #include<iostream>
02 using namespace std;
03 #define MAXN 20
04 int gu[MAXN][MAXN];
05 int luo(int n, int m) {
06     if(n <= 1 || m < 2)
07         return 1;
08     if(gu[n][m] != -1)
09         return gu[n][m];
10     int ans=0;
11     for(int i = 0; i <m; i += 2)
12         ans+=luo(n-1, i);
13     gu[n][m] = ans;
14     return ans;
15 }
16 int main() {
17     int n, m;
18     cin >> n >> m;
19     for(int i = 0; i < MAXN; i++)
20         for(int j = 0; j < MAXN; j++)
21             gu[i][j]= -1;
22     cout << luo(n, m);
23     return 0;
24 }
```

判断题

16. (1分) luo函数中，m的值不可能是奇数。（ ）

17. (1分) 若将第11行的“<”改为“<=”，程序的输出结果可能会改变。（ ）

18. (2分) 若将第8、9、13行删除，程序的运行的结果不变。（ ）

19. (2分) 在添加合适的头文件后，将第19到21行替换为”memset(gu, 255,sizeof(gu));“可以起到相同的作用。（ ）

单选题

20. (4分) 若输入数据为 4 8，则输出为（ ）。

A. 7　　B. 8　　C. 15　　D. 16

21. (3分) 最坏情况下，此程序的时间复杂度是（ ）。

A. O(m²n)　　B. O(nm!)　　C. O(n²)　　D. O(n²m)

(2)

```
01 #include<iostream>
02 using namespace std;
03 int n,m,i,j,p,k;
04 int a[100],b[100];
05 int main()
06 {
07     cin>>n>>m;
08     a[0]=n;i=0;p=0;k=1;
09     do{
10         for(j=0;j<i;j++)
11             if(a[i]==a[j])
12             {
13                 p=1;k=j;break;
14             }
15         if(p) break;
16         b[i]=a[i]/m;
17         a[i+1]=a[i]%m*10;
18         i++;
19     }while(a[i]!=0);
20     cout<<b[0]<<".";
21     for(j=1;j<k;j++) cout<<b[j];
22     if(p) cout<<"(";
23     for(j=k;j<i;j++) cout<<b[j];
24     if(p) cout<<") ";
25     cout<<endl;
26     return 0;
27 }
```

判断题

22. (1.5分) 程序输入的n和m不能相等。（ ）

23. (1.5分) 程序输入的m不能等于0。（ ）

24. (1.5分) 第9~19行的do…while循环一共有2个出口。（ ）

25. (1.5分) 数组a和b中的数值都小于或等于n。（ ）

单选题

26. (3分) 若输入数据为“11 8”，则输出结果为（ ）

A. 0.375　　B. 0.(375)　　C. 1.375　　D. 1.(375)

27. (3分) 若输入数据为“5 13”，则输出结果为（ ）。

A. 0.386514　　B. 0.(386514)　　C. 0.384615　　D. 0.(384615)

(3)

```
#include <algorithm>
#include <iostream>
using namespace std;

int n;
int d[50][2];
int ans;

void dfs(int n, int sum) {
    if (n == 1) {
        ans = max(sum, ans);
        return;
    }
    for (int i = 1; i < n; ++i) {
        int a = d[i - 1][0], b = d[i - 1][1];
        int x = d[i][0], y = d[i][1];
        d[i - 1][0] = a + x;
        d[i - 1][1] = b + y;
        for (int j = i; j < n - 1; ++j)
            d[j][0] = d[j + 1][0], d[j][1] = d[j + 1][1];
        int s = a + x + abs(b - y);
        dfs(n - 1, sum + s);
        for (int j = n - 1; j > i; --j)
            d[j][0] = d[j - 1][0], d[j][1] = d[j - 1][1];
        d[i - 1][0] = a, d[i - 1][1] = b;
        d[i][0] = x, d[i][1] = y;
    }
}

int main() {
    cin >> n;
    for (int i = 0; i < n; ++i)
        cin >> d[i][0];
```

```
34      for (int i = 0; i < n;++i)
35          cin >> d[i][1];
36      ans = 0;
37      dfs(n, 0);
38      cout << ans << endl;
39      return 0;
40  }
```

假设输入的 n 是不超过50的正整数，$d[i][0], d[i][i]$ 都是不超过10000的正整数，完成下面的判断题和单选题

判断题

28. (1.5分) 若输入 n 为 0，此程序可能会死循环或发生运行错误。（ ）

29. (1.5分) 若输入 n 为20,接下来的输入全为0，则输出为0。（ ）

30. (1.5分) 输出的数一定不小于输入的 $d[i][0]$ 和 $d[i][1]$ 的任意一个。（ ）

单选题

31. (3分) 若输入的 n 为20,接下来的输入是20个9和20个0,则输出为（ ）。

A. 1890　　B. 1881　　C. 1908　　D. 1917

32. (3分) 若输入的 n 为30,接下来的输入是30个0和30个5,则输出为（ ）。

A. 2000　　B. 2010　　C. 2030　　D. 2020

33. (4分) 若输入的 n 为15，接下来的输入是15到1，以及15到1，则输出为（ ）。

A. 2440　　B. 2220　　C. 2240　　D. 2420

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **9** 小题，共计 **26** 分，各小题分值见题目）

(1)

完善程序： (数字删除)下面程序的功能是将字符串中的数字字符删除后输出。请填空。

```
01  #include <iostream>
02  using namespace std;
03  int delnum( char *s )
04  {
05      int i, j;
06      j = 0;
07      for ( i = 0; s[i] != '\0'; i++ )
08          if ( s[i] < '0'   __(1)__  s[i] > '9' )
```

```
        {
            s[j] = s[i];
            __(2)__;
        }
    return(__(3)__);
}
const int SIZE = 30;
int main()
{
    char s[SIZE];
    int len, i;
    cin.getline( s, sizeof(s) );
    len = delnum( s );
    for ( i = 0; i < len; i++ )
        cout << __(4)__;
    cout << endl;
    return 0;
}
```

34. (3分) (1)处应填：_____

35. (3分) (2)处应填：_____

36. (3分) (3)处应填：_____

37. (3分) (4)处应填：_____

(2)

对于一个1到 n 的排列 P （即1到 n 中每一个数在 P 中出现了恰好一次），令 $q[i]=$第i 个位置之后第一个比 $P[i]$ 值更大的位置，如果不存在这样的位置，则$q[i]=n+1$ 。举例来说，如果 $n=5$ 且 P 为15423，则 q 为26656。

```
#include <iostream>
using namespace std;

const int N = 100010;
int n;
int L[N], R[N], a[N];

int main() {
    cin >> n;
    for (int i = 1; i <= n; ++i) {
        int x;
        cin >> x;
```

```
13          __(1)__ ;
14      }
15
16      for (int i = 1; i <= n; ++i) {
17          R[i] = __(2)__ ;
18          L[i] = i - 1;
19      }
20
21      for (int i = 1; i <= n; ++i) {
22          L[__(3)__] = L[a[i]];
23          R[L[a[i]]] = R[__(4)__];
24      }
25
26      for (int i = 1; i <= n; ++i) {
27          cout << __(5)__ << " ";
28      }
29
30      cout << endl;
31      return 0;
32  }
```

38. (3分) (1)处应填：_____

39. (2分) (2)处应填：_____

40. (3分) (3)处应填：_____

41. (3分) (4)处应填：_____

42. (3分) (5)处应填：_____

CSP-J 初赛模拟卷（第 07 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

1．计算机应用的最早领域（　）

A．数值计算　　B．人工智能　　C．机器人　　D．过程控制

2．二进制数11 1011 1001 0111和01 0110 1110 1011进行逻辑与运算的结果是（　）。

A．01 0010 1000 1011　　B．01 0010 1001 0011

C．01 0010 1000 0001　　D．01 0010 1000 0011

3．在内存储器中每个存储单元都被赋予一个唯一的序号，称为（　）

A．地址　　B．序号　　C．下标　　D．编号

4．如果开始时计算机处于小写输入状态，现在有一只小老鼠反复按照CapsLock、 字母键A、字母键 S、字母键D、字母键 F 的顺序循环按键，即 CapsLock、A、 S、D、F、CapsLock、A、S、D、F、……，屏幕上输出的第 81 个字符是字母（　）。

A．A　　B．S　　C．D　　D．a

5．分辨率为 800x600、16 位色的位图，存储图像信息所需的空间为（　）。

A．937.5KB　　B．4218.75KB　　C．4320KB　　D．2880KB

6．以比较作为基本运算，在N个数中找出最大数，最坏情况下所需要的最少的比较次数为（　）。

A．N^2　　B．N　　C．N-1　　D．N+1

7．以下是 32 位机器和 64 位机器的区别是（　）。

A．显示器不同　　B．硬盘大小不同　　C．寻址空间不同　　D．输入法不同

8．由四个没有区别的点构成的简单无向连通图的个数是（　）。

A．6　　B．7　　C．8　　D．9

9. 言泡排序算法的伪代码如下：

```
输入：数组L,n≥k。输出：按非递减顺序排序的L。
算法 BubbleSort:
    1. FLAG ← n //标记被交换的最后元素位置
    2. while FLAG > 1 do
    3. k ← FLAG -1
    4. FLAG ← 1
    5. for j=1 to k do
    6. if L(j) > L(j+1) then do
    7. L(j) ← L(j+1)
    8. FLAG ← j
```

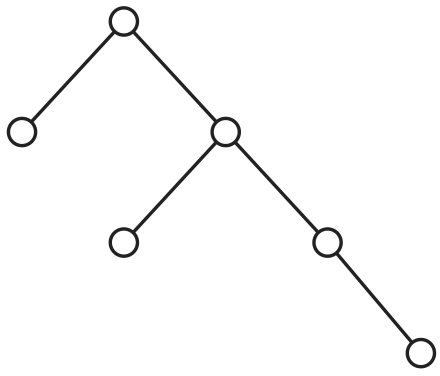
对 n 个数用以上音泡排序算法进行排序，最少需要比较多少次？（ ）。

A. n^2　　B. $n-2$　　C. $n-1$　　D. n

10. 表达式a*(b+c)*d 的后缀表达式为（ ），其中“*”和“+”是运算符。

A. **a+bcd　　B. abc+*d*　　C. abc+d**　　D. *a*+bcd

11. 一棵二叉例如右图所示，若采用顺序存储结构，即用一维数组元素存储该二叉树中的结点（根结点的下标为1，若某结点的下标为i，则其在孩子位于下标2i处、右孩子位于下标2i+1处），则该数组的最大下标至少为（ ）。



A. 6　　B. 10　　C. 15　　D. 12

12. 十进制小数 13.375 对应的二进制数是（ ）。

A. 1101.011　　B. 1011.011　　C. 1101.101　　D. 1010.01

13. 一些数字可以颠倒过来看，例如0、1、8颠倒过来还是本身，6颠倒过来是9，9颠倒过来看还是6,其他数字颠倒过来都不构成数字。 类似的，一些多位数也可以颠倒过来看，比如106颠倒过来是901。假设某个城市的车牌只由5位数字组成，每一位都可以取0到9。 请问这个城市最多有多少个车牌倒过来恰好还是原来的车牌？（ ）

A. 60　　B. 125　　C. 75　　D. 100

14. 为了统计一个非负整数的二进制形式中 1 的个数，代码如下： int CountBit(int x) { int ret = 0; while (x) { ret++; _____; } return ret; } 则空格内要填入的语句是（ ）。

A. x >>= 1　　B. x &= x - 1　　C. x |= x >> 1　　D. x <<= 1

15. 下列说法正确的是（ ）。

A. CPU的主要任务是执行数据运算和程序控制
B. 存储器具有记忆能力，其中信息任何时候都不会丢失
C. 两个显示器屏幕尺寸相同，则它们的分辨率必定相同
D. 个人用户只能使用 Wifi 的方式连接到 Internet

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 √，错误填 ×；共 18 小题，共计 39 分，各小题分值见题目）

(1)

```
01 #include <cstdio>
02 using namespace std;
03 int n;
04 int a[100];
05
06 int main() {
07     scanf("%d", &n);
08     for (int i = 1; i <= n; ++i)
09         scanf("%d", &a[i]);
10     int ans = 1;
11     for (int i = 1; i <= n; ++i) {
12         if (i > 1 && a[i] < a[i - 1])
13             ans = i;
14         while (ans < n && a[i] >= a[ans + 1])
15             ++ans;
```

```
16          printf("%d\n", ans);
17      }
18      return 0;
19  }
```

判断题

16. (1分) 第16行输出ans时，ans的值一定大于i。（ ）

17. (1分) 程序输出的ans小于等于n。（ ）

18. (1.5分) 若将第12行的 $<$ 改为 $!=$ ，程序输出的结果不会改变。（ ）

19. (1.5分) 当程序执行到第16行时，若ans-i>2,则 $a[i+1] \leq a[i]$ 。（ ）

单选题

20. (3分) 若输入的a数组是一个严格单调递增的数列，此程序的时间复杂度（ ）

A. O(log n)　　B. O(n^2)　　C. O(n log n)　　D. O(n)

21. (4分) 最坏情况下，此程序的时间复杂度是（ ）

A. O(n^2)　　B. O(log n)　　C. O(n)　　D. O(n log n)

(2)

```
01  #include <stdio.h>
02  #include <string.h>
03
04  char base[64];
05  char table[256];
06  char str[256];
07  char ans[256];
08
09  void init()
10  {
11      for (int i = 0; i < 26; i++) base[i] = 'A' + i;
12      for (int i = 0; i < 26; i++) base[26 + i] = 'a' + i;
13      for (int i = 0; i < 10; i++) base[52 + i] = '0' + i;
14      base[62] = '+', base[63] = '/';
15
16      for (int i = 0; i < 256; i++) table[i] = 0xff;
17      for (int i = 0; i < 64; i++) table[base[i]] = i;
18      table['='] = 0;
19  }
```

```
20
21 void decode(char *str)
22 {
23     char *ret = ans;
24     int i, len = strlen(str);
25     for (i = 0; i < len; i += 4) {
26         (*ret++) = table[str[i]] << 2 | table[str[i + 1]] >> 4;
27         if (str[i + 2] != '=')
28             (*ret++) = (table[str[i + 1]] & 0x0f) << 4 | table[str[i
   + 2]] >> 2;
29         if(str[i + 3] != '=')
30             (*ret++) = table[str[i + 2]] << 6 | table[str[i + 3]];
31     }
32 }
33
34 int main()
35 {
36     init();
37     printf("%d\n", (int)table[0]);
38
39     scanf("%s", str);
40     decode(str);
41     printf("%s\n", ans);
42     return 0;
43 }
```

假设 char 是 8 位有符号类型，输入字符串长度为 4 的倍数，且输入不超过数组范围。

判断题

22. (1.5分) 输出的第二行一定是由小写字母、大写字母、数字和 +、/、= 构成的字符串。（ ）

23. (1.5分) 可能存在输入不同，但输出的第二行相同的情形。（ ）

24. (1.5分) 输出的第一行为 -1。（ ）

单选题

25. (3分) 设输入字符串长度为 n，decode 函数的时间复杂度为（ ）。

A. $O(\sqrt{n})$　　B. $O(n)$　　C. $O(n\log n)$　　D. $O(n^2)$

26. (3分) 当输入为 Y3Nx 时，输出的第二行为（ ）。

A. csp　　B. csq　　C. CSP　　D. Csp

27. (3.5分) 当输入为 Y2NmIDIwMjE= 时，输出的第二行为（ ）。

A. ccf2021　　B. ccf2022　　C. ccf 2021　　D. ccf 2022

(3)

```
#include<cstdio>
#include<cstring>
using namespace std;
int n, m;
int f[101][101];
int F[101][101];
int main(){
    scanf("%d%d",&n,&m);//n的值在1到100之间
    memset(f,-1,sizeof(f));
    for(int i = 1; i <=  m; i++) {
        int u,v,w;//w的值在0到10000之间
        scanf("%d%d%d",&u,&v,&w);
        f[u][v]=f[v][u]=w;
    }
    for(int k = 1; k <= n; k++)
        for(int i=1;i <=n; i++)
            for(int j=1;j<= n; j++)
                if(f[i][k]!=-1&&f[k][j]!=-1)
                    if(f[i][j]==-1||f[i][j]>f[k][j]+f[i][k])
                        f[i][j]=f[i][k]+f[k][j];
    int ans=2147483647;
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n; j++){
            for(int x=1;x<=n; x++)
                for(int y=1;y<=n; y++)
                    F[x][y]=f[x][y];
            F[i][j]=F[j][i]=0;
            for(int x=1;x<= n; x++)
                for(int y=1;y<=n; y++)
                    if(F[x][y]==-1||F[x][y]>F[x][i]+F[i][y])
                        F[x][y]=F[x][i]+F[i][y];
            for(int x=1;x<=n; x++)
                for(int y=1;y<=n; y++)
                    if(F[x][y]==-1||F[x][y]>F[x][j]+F[j][y])
                        F[x][y]=F[x][j]+F[j][y];
```

```
36              int res=0;
37              for(int x=1;x<=n; x++)
38                  for(int y=1;y<x; y++)
39                      res +=F[x][y];
40              ans = res < ans ? res : ans;
41          }
42      printf("%d\n",ans);
43      return 0;
44  }
```

- 判断题

判断题

28. (1分) 14到16行，将外层到内层的循环变量依次调整为i、j、k，程序的运行的结果不变。（ ）

29. (2分) 这个程序的时间复杂度和m无关。（ ）

30. (2分) 20行的ans如果初始化为10^7时，可能无法得到正确结果。（ ）

31. (2分) 若将第27到30行的部分和31到34行的两个部分互换，程序的运行的结果不变。（ ）

单选题

32. (3分) 若输入数据为4 5/1 2 3/1 3 6/2 3 4/2 4 7/3 4 2(其中“/”为换行符)，则输出为（ ）。

A. 14　　B. 18　　C. 21　　D. 28

33. (3分) 这个程序使用了（ ）算法。

A. Floyd　　B. Dijkstra　　C. Prim　　D. Kruskal

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **10** 小题，共计 **29** 分，各小题分值见题目）

(1)

（计数排序）计数排序是一个广泛使用的排序方法。下面的程序使用双关键字计数排序，将 n 对10000以内的整数，从小到大排序。

例如有三对整数(3,4)、(2,4)、(3,3),那么排序之后应该是(2,4)、(3,3)、(3,4)。

输入第一行为n,接下来n行，第i行有两个数 $a[i]$ 和 $b[i]$ ，分别表示第i对整数的第一关键字和第二关键字。

从小到大排序后输出。

数据范围 $1 < n < 10^7, 1 < a[x], b[x] < 10^4$。提示:应先对第二关键字排序，再对第一关键字排序。数组ord[]存储第二关键字排序的结果，数组res[]存储双关键字排序的结果。试补全程序。

```
01 #include <cstdio>
02 #include <cstring>
03 using namespace std;
04 const int maxn = 10000000;
05 const int maxs = 10000;
06
07 int n;
08 unsigned a[maxn], b[maxn],res[maxn], ord[maxn];
09 unsigned cnt[maxs + 1];
10 int main() {
11     scanf("%d", &n);
12     for (int i = 0; i < n; ++i)
13         scanf("%u%u", &a[i], &b[i]);
14     memset(cnt, 0, sizeof(cnt));
15     for (int i = 0; i < n; ++i)
16         __(1)__; // 利用 cnt 数组统计数量
17     for (int i = 0; i < maxs; ++i)
18         cnt[i + 1] += cnt[i];
19     for (int i = 0; i < n; ++i)
20         __(2)__; // 记录初步排序结果
21     memset(cnt, 0, sizeof(cnt));
22     for (int i = 0; i < n; ++i)
23         __(3)__; // 利用 cnt 数组统计数量
24     for (int i = 0; i < maxs; ++i)
25         cnt[i + 1] += cnt[i];
26     for (int i = n - 1; i >= 0; --i)
27         __(4)__; // 记录最终排序结果
28     for (int i = 0; i < n; i++)
29         printf("%u %u\n", __(5)__);
30
31     return 0;
32 }
```

34. (3分) (1)处应填

A. ++cnt [i]　　B. ++cnt[b[i]]
C. ++cnt[a[i] * maxs + b[i]]　　D. ++cnt[a[i]]

35. (3分) (2)处应填

A. ord[--cnt[a[i]]] = i　　B. ord[--cnt[b[i]]] = a[i]
C. ord[--cnt[a[i]]] = b[i]　　D. ord[--cnt[b[i]]] = i

36. (3分) (3)处应填

A. ++cnt[b[i]]　　B. ++cnt[a[i] * maxs + b[i]]
C. ++cnt[a[i]]　　D. ++cnt[i]

37. (3分) (4)处应填

A. res[--cnt[a[ord[i]]]] = ord[i]　　B. res[--cnt[b[ord[i]]]] = ord[i]
C. res[--cnt[b[i]]] = ord[i]　　D. res[--cnt[a[i]]] = ord[i]

38. (3分) (5)处应填

A. a[i], b[i]　　B. a[res[i]], b[res[i]]
C. a[ord[res[i]]], b[ord[res[i]]]　　D. a[res[ord[i]]], b[res[ord[i]]]

(2)

完善程序

（郊游活动）有n名同学参加学校组织的郊游活动，已知学校给这n名同学的郊游总结题为A元，与此同时第i位同学自己携带了M元。为了方便郊游，活动地点提供 $B(\geq n)$ 钢游行车供人租用，租用第j辆自行车的价格为C1元，每位同学可以使用自己携带的钱或者学校的郊游经验，为了方便账务管理，每位同学只能为自己租用自行车，且不会借钱给他人，他们供图道最多有多少位同学能够租用到自行车。

本题采用二分法。对于区间 $[l,r]$ ，我们取中间点 mid 并判断租用到自行车的人数能否达到mid。判断的过程是利用贪心算法实现的。

```
01 #include <iostream>
02 using namespace std;
03 #define MAXN 1000000
04 int n, B, A, M[MAXN], C[MAXN], l, r, ans, mid;
05 bool check(int nn) {
06     int count = 0, i, j;
07     i = __(1)__;
```

```
    j = 1;
    while (i <= n) {
        if(__(2)__)
            count += C[j] - M[i];
        i++;
        j++;
    }
    return __(3)__;
}
void sort(int a[], int l, int r) {
    int i = l, j = r, x = a[(l + r) / 2], y;
    while (i <= j) {
        while (a[i] < x) i++;
        while (a[j] > x) j--;
        if (i <= j) {
            y = a[i]; a[i] = a[j]; a[j] = y;
            i++; j--;
        }
    }
    if (i < r) sort(a, i, r);
    if (l < j) sort(a, l, j);
}
int main() {
    int i;
    cin >> n >> B >> A;
    for (i = 1; i <= n; i++)
        cin >> M[i];
    for (i = 1; i <= B; i++)
        cin >> C[i];
    sort(M, 1, n);
    sort(C, 1, B);
    l = 0;
    r = n;
    while (l <= r) {
        mid = (l + r) / 2;
        if(__(4)__){
            ans = mid;
            l = mid + 1;
        }else
            r = __(5)__;
```

```
    }
    cout << ans << endl;
    return 0;
}
```

39. (3分) (1)处应填：_____

40. (3分) (2)处应填：_____

41. (3分) (3)处应填：_____

42. (2.5分) (4)处应填：_____

43. (2.5分) (5)处应填：_____

CSP-J 初赛模拟卷（第 08 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

1．以下不属于无线通信技术的是（　）。

A．蓝牙　　B．Wifi　　C．GPRS　　D．以太网

2．在 8 位二进制补码中，10101011 表示的数是十进制下的（　）。

A．43　　B．-85　　C．-43　　D．-84

3．以下不是存储设备的是（　）。

A．光盘　　B．磁盘　　C．固态硬盘　　D．鼠标

4．编译器的主要功能（　）

A．将源程序翻译成机器指令代码　　B．将源程序重新组合

C．将低级语言翻译成高级语言　　D．将一种高级语言翻译成另一种高级语言

5．若有如下程序段,其中s、a、b、c均已定义为整型变量,且a、c均已赋值(c大于0)

```
s = a;
for(b = 1;b<=c;b++)
s = s - 1;
```

则与上述程序段功能等价的赋值语句是（　）

A．$s=a-c$;　　B．$s=a-b$;　　C．s=s-c;　　D．s=b-c;

6．给定一个含 N 个不相同数字的数组，在最坏情况下，找出其中最大或最小的数，至少需要 $N-1$ 次比较操作。则最坏情况下，在该数组中同时找最大与最小的数至少需要（　）次比较操作。（[表示向上取整，]表示向下取整）

A．$\lceil 3N/2 \rceil - 2$　　B．$\lfloor 3N/2 \rfloor - 2$　　C．$2N-2$　　D．$2N-4$

7．由1，1，2，2，3这五个数字组成不同的三位数有（　）种。

A．18　　B．15　　C．12　　D．24

8．向一个栈顶指针为 hs 的链式栈中插入一个指针s 指向的结点时，应执行（　）。

A．hs->next = s;　　B．s->next = hs; hs = s;

C．s->next = hs->next; hs->next = s;　　D．s->next = hs; hs = hs->next;

9．对于有n个顶点、m条边的无向联通图（m>n），需要删掉（　）条边才能使其成为一棵树。

A．n-1　　B．m-n　　C．m-n-1　　D．m-n+1

10．把8个同样的球放在5个同样的袋子里，允许有的袋子空着不放，问共有多少种不同的分法？（　） 提示：如果8个球都放在一个袋子里，无论是哪个袋子，都只算同一种分法。

A．22　　B．24　　C．18　　D．20

11．独根树的高度为 1,具有 61 个结点的完全二叉树的高度为（　）。

A．5　　B．6　　C．7　　D．8

12．10000 以内，与 10000 互质的正整数有（　）个。

A．2000　　B．4000　　C．6000　　D．8000

13．有以下程序：

```
#include <iostream>
using namespace std;
int main()
{
    int k = 4, n = 0;
    while (n < k)
    {
        n++;
        if (n % 3 != 0)
            continue;
        k--;
    }
    cout << k << "," << n << endl;
    return 0;
}
```

程序运行后输出的结果是（ ）。

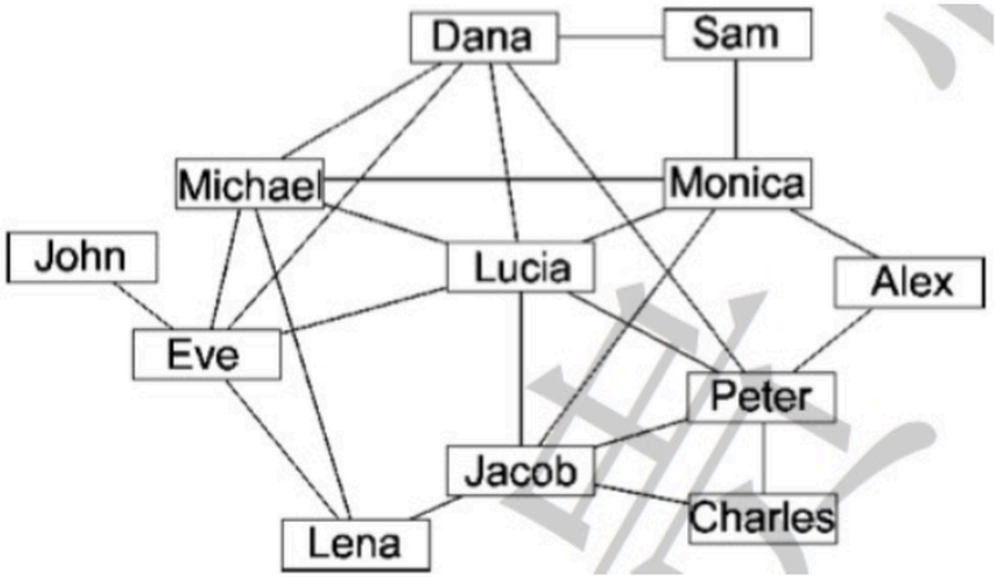
A．2,2　　B．2,3　　C．3,2　　D．3,3

14．操作系统的作用是（　）。

A．把源程序译成目标程序　　B．便于进行数据管理

C．控制和管理系统资源　　D．实现硬件之间的连接

15. Lucia 和她的朋友以及朋友的朋友都在某社交网站上注册了账号。下图是他 们之间的关系图，两个人之间有边相连代表这两个人是朋友，没有边相连代表不是朋友。这个社交网站的规则是：如果某人 A 向他（她）的朋友 B 分享了某张照片，那么 B 就可以对该照片进行评论；如果 B 评论了该照片，那 么他（她）的所有朋友都可以看见这个评论以及被评论的照片，但是不能对该照片进行评论（除非 A 也向他（她）分享了该照片）。现在 Lucia 已经上传了一张照片，但是她不想让 Jacob 看见这张照片，那么她可以向以下朋友（ ）分享该照片。



A. Dana, Michael, Eve　　B. Dana, Eve, Monica

C. Michael, Eve, Jacob　　D. Micheal, Peter, Monica

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 **√**，错误填 **×**；共 **18** 小题，共计 **41** 分，各小题分值见题目）

(1)

```
01 #include <iostream>
02 #include <cmath>
03 using namespace std;
04
05 const double r = acos(0.5);
06
07 int a1, b1, c1, d1;
08 int a2, b2, c2, d2;
09
10 inline int sq(const int x) { return x * x; }
11 inline int cu(const int x) { return x * x * x; }
```

```
12
13 int main()
14 {
15     cout.flags(ios::fixed);
16     cout.precision(4);
17
18     cin >> a1 >> b1 >> c1 >> d1;
19     cin >> a2 >> b2 >> c2 >> d2;
20
21     int t = sq(a1 - a2) + sq(b1 - b2) + sq(c1 - c2);
22
23     if (t <= sq(d2 - d1)) cout << cu(min(d1, d2)) * r * 4;
24     else if (t >= sq(d2 + d1)) cout << 0;
25     else {
26         double x = d1 - (sq(d1) - sq(d2) + t) / sqrt(t) / 2;
27         double y = d2 - (sq(d2) - sq(d1) + t) / sqrt(t) / 2;
28         cout << (x * x * (3 * d1 - x) + y * y * (3 * d2 - y)) * r;
29     }
30     cout << endl;
31     return 0;
32 }
```

假设输入的所有数的绝对值都不超过 1000，完成下面的判断题和单选题：

判断题

16. (1.5分) 将第 21 行中 t 的类型声明从 int 改为 double，不会影响程序运行的结果。（ ）

17. (1.5分) 将第 26、27 行中的“/sqrt(t)/2”替换为“/2 /sqrt(t)”，不会影响程序运行的结果。（ ）

18. (1.5分) 将第 28 行中的“x * x”改成“sq(x)”、“y * y”改成“sq(y)”，不会影响程序运行的结果。（ ）

19. (1.5分) 当输入为“0 0 0 1 1 0 0 1”时，输出为“1.3090”。（ ）

单选题

20. (3分) 当输入为“1 1 1 1 1 1 1 2”时，输出为（ ）。

A. “3.1416”　　B. “6.2832”　　C. “4.7124”　　D. “4.1888”

21. (3分) 这段代码的含义为（ ）

A. 求圆的面积并　B. 求球的体积并　C. 求球的体积交　D. 求椭球的体积并

(2)

```
#include <iostream>
using namespace std;

long long n, ans;
int k, len;
long long d[1000000];

int main() {
    cin >> n >> k;
    d[0] = 0;
    len= 1;
    ans = 0;
    for (long long i = 0; i <n; ++i) {
        ++d[0];
        for (int j = 0; j + 1<len; ++j) {
            if (d[j] == k) {
                d[j] = 0;
                d[j + 1] += 1;
                ++ans;
            }
        }
        if (d[len - 1] == k) {
            d[len - 1] = 0;
            d[len] =1;
            ++len;
            ++ans;
        }
    }
    cout << ans << endl;
    return 0;
}
```

假设输入的 n 是不超过 2^{62} 的正整数，k 是不超过 10000 的正整数，完成下面的判断题和单选题。

判断题

22. (1.5分) 若 $k=1$，则输出 ans 时，len = n。（ ）

23. (1.5分) 若 $k > 1$，则输出 ans 时，len 一定小于 n。（ ）

24. (1.5分) 若 $k > 1$，则输出 ans 时，k^{len} 一定大于 n。（ ）

单选题

25. (3分) 若输入的 $n = 10^{15}$，$k = 1$，则输出等于（ ）。

A. 1　　B. $(10^{30} - 10^{15})/2$　　C. $(10^{30} + 10^{15})/2$　　D. 10^{15}

26. (3分) 若输入的 $n = 205891132094649$（即 3^{30}），$k = 3$，则输出等于（ ）。

A. 3^{30}　　B. $(3^{30} - 1)/2$　　C. $3^{30} - 1$　　D. $(3^{30} + 1)/2$

27. (3分) 若输入的 $n = 100010002000090$，$k = 10$，则输出等于（ ）。

A. 11112222444543　B. 11122222444453
C. 11122222444543　D. 11112222444453

(3)

```
01 #include <stdio.h>
02
03 #define n 100000
04 #define N n+1
05
06 int m;
07 int a[N], b[N], c[N], d[N];
08 int f[N], g[N];
09
10 void init()
11 {
12     f[1] = g[1] = 1;
13     for (int i = 2; i <= n; i++) {
14         if (!a[i]) {
15             b[m++] = i;
16             c[i] = 1, f[i] = 2;
17             d[i] = 1, g[i] = i + 1;
18         }
19         for (int j = 0; j < m && b[j] * i <= n; j++) {
20             int k = b[j];
21             a[i * k] = 1;
22             if (i % k == 0) {
23                 c[i * k] = c[i] + 1;
24                 f[i * k] = f[i] / c[i * k] * (c[i * k] + 1);
25                 d[i * k] = d[i];
```

```
26                  g[i * k] = g[i] * k + d[i];
27                  break;
28              }
29              else {
30                  c[i * k] = 1;
31                  f[i * k] = 2 * f[i];
32                  d[i * k] = g[i];
33                  g[i * k] = g[i] * (k + 1);
34              }
35          }
36      }
37  }
38
39  int main()
40  {
41      init();
42
43      int x;
44      scanf("%d", &x);
45      printf("%d %d\n", f[x], g[x]);
46      return 0;
47  }
```

假设输入的x是不超过1000的自然数，完成下面的判断题和单选题:

判断题

28. (1.5分) 若输入不为"1"，把第12行删去不会影响输出的结果。（ ）

29. (2分) 第24行 $f[i*k]=f[i]/c[i*k]*(c[i*k]+1)$ 中的"$f[i]/c[i*k]$"可能存在无法整除而向下取整的情况。（ ）

30. (2分) 在执行完init()后，f数组不是单调递增的，但g数组是单调递增的。（ ）

单选题

31. (3分) init函数的时间复杂度为（ ）。

A. $O(n)$　　B. O(nlogn)　　C. O(n√n)　　D. O(n²)

32. (3分) 在执行完init()后，f[1], f[2], f[3]…..f[100]中有（ ）个等于2。

A. 23　　B. 24　　C. 25　　D. 26

33. (4分) 当输入”1000”时，输出为（ ）。

A. “15 1340”　　B. “15 2340”　　C. “16 2340”　　D. “16 1340”

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **10** 小题，共计 **29** 分，各小题分值见题目）

(1)

对于一个1到 n 的排列 P （即1到 n 中每一个数在 P 中出现了恰好一次），令 $q[i]$ 为第 i 个位置之后第一个比 $P[i]$ 值更大的位置，如果不存在这样的位置，则$q[i]=n+1$ 。举例来说，如果 $n=5$ 且 P 为15423，则 q 为26656。

下列程序读入了排列 P ，使用双向链表求解了答案。试补全程序。

```
01 #include <iostream>
02 using namespace std;
03 const int N = 100010;
04 int n;
05 int L[N], R[N], a[N];
06 int main() {
07     cin >> n;
08     for (int i = 1; i <= n; ++i) {
09         int x;
10         cin >> x;
11         __(1)__ ;
12     }
13     for (int i = 1; i <= n; ++i) {
14         R[i] = __(2)__ ;
15         L[i] = i - 1;
16     }
17     for (int i = 1; i <= n; ++i) {
18         L[__(3)__] = L[a[i]];
19         R[L[a[i]]] = R[__(4)__];
20     }
21     for (int i = 1; i <= n; ++i) {
22         cout << __(5)__ << " ";
23     }
24     cout << endl;
25     return 0;
26 }
```

34. (3分) (1)处应填：_____

35．(2分) (2)处应填：_____

36．(3分) (3)处应填：_____

37．(3分) (4)处应填：_____

38．(3分) (5)处应填：_____

(2)

2(烧作业)某课作业布置了N(3sNs10000)个题目，第题对应的得分是a。作业的总得分的计算方式为去掉作业中得分最小的一个题，剩下其它所有题目得分的平均值。但很不幸小A遇到了一场火灾，前K(1+K+N-2)个题目被烧了，无法记录得分。小A想知道，K是多少时，可以得到最高的作业得分？作业被烧了前K页，这时的得分是从第K+1页到最后一页中，去除最小得分后取平均值。

输入第一行是整数N，第二行是n个不超过10000的非负整数表示a

输出一行，若干个整数表示答案。如果有多个K，请依次升序输出。

```
01 #include<cstdio>
02 #include<cmath>
03 #define min(a,b)(a<b?a:b)
04 #define MAXN 100002
05 using namespace std;
06 int n, k[MAXN], cnt=0;
07 int s[MAXN], minScore, sum;
08 double maxAverage=0,nowAverage;
09 int main() {
10     scanf("%d",&n);
11     for(int i=1;i<= n;i++)
12         scanf("%d",&s[i]);
13     minScore=s[n];
14     __(1)__;
15     for(int i=n-1;i>=2;i--) {
16         minScore=min(minScore,s[i]);
17             __(2)__;
18         nowAverage=__(3)__;
19         if(nowAverage> maxAverage){
20             __(4)__
21                 maxAverage=nowAverage;
22         } else if(fabs(nowAverage-maxAverage)<1e-6)
23             __(5)__;
```

```
    }
    for(int i=cnt; i >= 1; i--)
        printf("%d\n",k[i]);
    return 0;
}
```

39. (3分) (1)处应填

A. sum=n　　B. sum=s[1]　　C. sum=s[n]　　D. sum=0

40. (3分) (2)处应填

A. sum=maxAverage*(n-i)　　B. sum+=s[i]
C. sum+=s[n-i]　　D. sum=s[i]+minScore

41. (3分) (3)处应填

A. (double)(sum+minScore)/(n-i)　　B. sum*1.0/(n-i)
C. (int)(sum-minScore)/(n-i)　　D. (double)(sum-minScore)/(n-i)

42. (3分) (4)处应填

A. k[++cnt]=i;　　B. k[cnt++]=i-1
C. cnt=1;k[cnt]=i-1;　　D. cnt=0;k[cnt]=l;

43. (3分) (5)处应填

A. k[cnt++]=i;　　B. k[++cnt]=i-1;　　C. k[cnt++]=n-i;　　D. k[cnt]=i;

CSP-J 初赛模拟卷（第 09 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

1．以下不属于面向对象程序设计语言的是（ ）。

A．C++　　B．Python　　C．Java　　D．C

2．在内存储器中每个存储单元都被赋予一个唯一的序号，称为（ ）

A．地址　　B．序号　　C．下标　　D．编号

3．广域网的英文缩写是（ ）。

A．LAN　　B．WAN　　C．MAN　　D．LNA

4．现有一张分辨率为 2048×1024 像素的 32 位真彩色图像。请问要存储这张图像，需要多大的存储空间？（ ）。

A．16MB　　B．4MB　　C．8MB　　D．2MB

5．对于入栈顺序为 a,b,c,d,e 的序列，下列（ ）不是合法的出栈序列。

A．a，b，c，d，e　　B．e，d，c，b，a

C．b，a，c，d，e　　D．c，d，a，e，b

6．设有100个已排好序的数据元素，采用折半查找时，最大比较次数为（ ）。

A．7　　B．10　　C．6　　D．8

7．有 10 个顶点的无向图至少应该有（ ）条边才有可能构成连通图。

A．9　　B．10　　C．11　　D．12

8．10 个三好学生名额分配到 7 个班级，每个班级至少有一个名额，一共有（ ）种不同的分配方案。

A．84　　B．72　　C．56　　D．504

9．独根树的高度为 1,具有 61 个结点的完全二叉树的高度为（ ）。

A．5　　B．6　　C．7　　D．8

10．把8个同样的球放在5个同样的袋子里，允许有的袋子空着不放，问共有多少种不同的分法？（　） 提示：如果8个球都放在一个袋子里，无论是哪个袋子，都只算同一种分法。

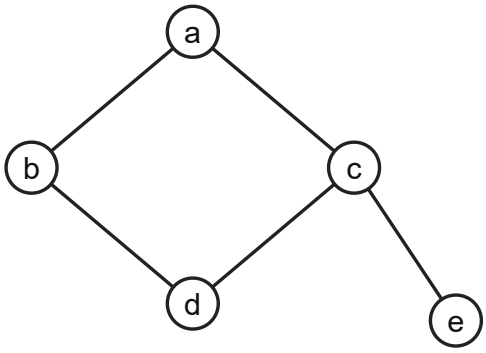
A．22　　B．24　　C．18　　D．20

11．在数据压缩编码中的哈夫曼编码方法，在本质是一种（　）的策略。

A．枚举　　B．贪心　　C．递归　　D．动态规划

12．考虑如下递归算法，则调用solve(7)得到的返回结果为（　）。

```
01 solve(n)
02 if n<=1 return 1
03 else if n>=5 return n*solve(n-2)
04 else return n*solve(n-1)
```



A．105　　B．840　　C．210　　D．420

13．以a为起点，对右边的无向图进行深度优先遍历，则b,c,d,e四个点中有可能作为最后一个遍历到的点个数为（　）。

A．1　　B．2　　C．3　　D．4

14．有四个人要从A点坐一条船过河到B点，船一开始在A点。该船一次最多可坐两个人。已知这四个人中每个人独自坐船的过河时间分别为1，2，4，8，且两个人坐船的过河时间为两人独自过河时间的较大者。则最短（　）时间可以让四个人都过河到B点（包括从B点把船开回A点时间）。

A．14　　B．15　　C．16　　D．17

15．319和377的最大公约数是（　）。

A．27　　B．33　　C．29　　D．31

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 **√**，错误填 **×**；共 **18** 小题，共计 **39.5** 分，各小题分值见题目）

(1)

```
01 #include <cstdio>
02 #include <cstring>
03 using namespace std;
04 char st[100001];
05 int main() {
06     scanf("%s", st);
07     int n = strlen(st);
08     for (int i = 1; i <= n; ++i) {
09         if (n % i == 0) {
10             char c = st[i - 1];
11             if (c >= 'a')
12                 st[i - 1] = c - 'a' + 'A';
13         }
14     }
15     printf("%s", st);
16     return 0;
17 }
```

判断题

16. (1.5分) 输入的字符串只能由小写字母或大写字母组成。（ ）

17. (1.5分) 若将第8行的“i=1”改为“i=0”，程序运行时会发生错误。（ ）

18. (1.5分) 若将第8行的“ $i <= n$ ”改为“ $i * i <= n$ ”，程序运行结果不会改变。（ ）

19. (1.5分) 若输入的字符串全部由大写字母组成，那么输出的字符串就跟输入的字符串一样。（ ）

单选题

20. (3分) 若输入的字符串长度为18,那么输入的字符串跟输出的字符串相比，至多有（ ）个字符不同。

A. 18　　B. 6　　C. 10　　D. 1

21. (3分) 若输入的字符串长度为（ ），那么输入的字符串跟输出的字符串相比，至多有36个字符不同。

A. 36　　B. 100000　　C. 1　　D. 128

(2)

```
01 #include <cstdio>
02 using namespace std;
03 int n;
04 int a[100];
05
06 int main() {
07     scanf("%d", &n);
08     for (int i = 1; i <= n; ++i)
09         scanf("%d", &a[i]);
10     int ans = 1;
11     for (int i = 1; i <= n; ++i) {
12         if (i > 1 && a[i] < a[i - 1])
13             ans = i;
14         while (ans < n && a[i] >= a[ans + 1])
15             ++ans;
16         printf("%d\n", ans);
17     }
18     return 0;
19 }
```

判断题

22. (1分) 第16行输出ans时，ans的值一定大于i。（ ）

23. (1分) 程序输出的ans小于等于n。（ ）

24. (1.5分) 若将第12行的 $<$ 改为 `!=` ，程序输出的结果不会改变。（ ）

25. (1.5分) 当程序执行到第16行时，若 $\mathrm{ans}-\mathrm{i}>2$，则 $\mathrm{a}[i+1]\le \mathrm{a}[i]$。（ ）

单选题

26. (3分) 若输入的a数组是一个严格单调递增的数列，此程序的时间复杂度 （ ）

A. O(log n)　　B. O(n^2)　　C. O(n log n)　　D. O(n)

27. (4分) 最坏情况下，此程序的时间复杂度是 （ ）

A. O(n^2)　　B. O(log n)　　C. O(n)　　D. O(n log n)

(3)

```
01 #include <stdio.h>
02
03 #define n 100000
04 #define N n+1
05
06 int m;
07 int a[N], b[N], c[N], d[N];
08 int f[N], g[N];
09
10 void init()
11 {
12     f[1] = g[1] = 1;
13     for (int i = 2; i <= n; i++) {
14         if (!a[i]) {
15             b[m++] = i;
16             c[i] = 1, f[i] = 2;
17             d[i] = 1, g[i] = i + 1;
18         }
19         for (int j = 0; j < m && b[j] * i <= n; j++) {
20             int k = b[j];
21             a[i * k] = 1;
22             if (i % k == 0) {
23                 c[i * k] = c[i] + 1;
24                 f[i * k] = f[i] / c[i * k] * (c[i * k] + 1);
25                 d[i * k] = d[i];
26                 g[i * k] = g[i] * k + d[i];
27                 break;
28             }
29             else {
30                 c[i * k] = 1;
31                 f[i * k] = 2 * f[i];
32                 d[i * k] = g[i];
33                 g[i * k] = g[i] * (k + 1);
34             }
35         }
36     }
37 }
38
39 int main()
40 {
41     init();
```

```
42
43      int x;
44      scanf("%d", &x);
45      printf("%d %d\n", f[x], g[x]);
46      return 0;
47  }
```

假设输入的x是不超过1000的自然数，完成下面的判断题和单选题：

判断题

28. (1.5分) 若输入不为“1”，把第12行删去不会影响输出的结果。（ ）

29. (2分) 第24行 $f[i*k]=f[i]/c[i*k]*(c[i*k]+1)$ 中的“$f[i]/c[i*k]$”可能存在无法整除而向下取整的情况。（ ）

30. (2分) 在执行完init()后，做组不是单调递增的，但g数组是单调递增的。（ ）

单选题

31. (3分) init函数的时间复杂度为（ ）。

A. O(n)　　B. O(nlogn)　　C. O(n√n)　　D. O(n²)

32. (3分) 在执行完init()后，f[1], f[2], f[3]…..f[100]中有（ ）个等于2。

A. 23　　B. 24　　C. 25　　D. 26

33. (4分) 当输入”1000”时，输出为（ ）。

A. “15 1340”　　B. “15 2340”　　C. “16 2340”　　D. “16 1340”

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **10** 小题，共计 **29** 分，各小题分值见题目）

(1)

完善程序：

（快速幂）请完善下面的程序，该程序使用分治法求 $x^p \mod m$ 的值。

输入:三个不超过 10000 的正整数 x, p, m。

输出：$x^p \mod m$ 的值。

提示:若 p 为偶数，$x^p = (x^2)^{p/2}$；若 p 为奇数，$x^p = x * (x^2)^{(p-1)/2}$。

```
01 #include<iostream>
02 using namespace std;
03 int x, p, m, i, result;
04 int main(){
05     cin >> x >> p >> m;
06     result = __(1)__;
07     while (__(2)__){
08         if (p % 2 == 1)
09             result = __(3)__;
10         p /= 2;
11         x = __(4)__;
12     }
13     cout << __(5)__ << endl;
14     return 0;
15 }
```

34. (2分) (1)处应填：_____

35. (3分) (2)处应填：_____

36. (3分) (3)处应填：_____

37. (3分) (4)处应填：_____

38. (3分) (5)处应填：_____

(2)

（Josephus 问题）有 n个人围成一个圈，依次标号 0 至 n-1。从 0 号开始，依次 0, 1, 0, 1, … 交替报数，报到 1 的人会离开，直至圈中只剩下一个人。求最后剩下人的编号。 试补全模拟程序。

```
01 #include <stdio.h>
02
03 const int MAXN = 1000000;
04 int F[MAXN];
05
06 int main(){
07     int n;
08     scanf("%d", &n);
09     int i = 0, p = 0, c = 0;
10     while( __(1)__){
```

```
11          if (F[i] == 0){
12              if ( __(2)__ ){
13                  F[i] = 1;
14                  __(3)__;
15              }
16              __(4)__;
17          }
18          __(5)__;
19      }
20      int ans = -1;
21      for (int i = 0; i < n; i++)
22          if (F[i] == 0)
23              ans = i;
24      printf("%d\n", ans);
25      return 0;
26  }
```

39. (3分) (1)处应填

A. i < n　　B. c < n　　C. i < n - 1　　D. c < n - 1

40. (3分) (2)处应填

A. i % 2 == 0　　B. i % 2 == 1　　C. p　　D. !p

41. (3分) (3)处应填

A. i++　B. i = (i+1) % n　C. c++　D. p ^= 1

42. (3分) (4)处应填

A. i++　B. i = (i+1) % n　C. c++　D. p ^= 1

43. (3分) (5)处应填

A. i++　B. i = (i+1) % n　C. c++　D. p ^= 1

CSP-J 初赛模拟卷（第 10 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

1．中国计算机学会于（　）年创办全国青少年计算机程序设计竞赛。

A．1983　　B．1984　　C．1985　　D．1986

2．目前主流的计算机储存数据最终都是转换成（　）数据进行储存。

A．二进制　　B．十进制　　C．八进制　　D．十六进制

3．以下奖项与计算机领域最相关的是（　）。

A．奥斯卡奖　　B．图灵奖　　C．诺贝尔奖　　D．普利策奖

4．下列四个不同进制的数中，与其它三项数值上不相等的是

A．(269)16　B．(617)10　C．(1151)8　D．(1001101011)2

5．链表不具有的特点是（　）。

A．可随机访问任一元素　　B．不必事先估计存储空间

C．插入删除不需要移动元素　D．所需空间与线性表长度成正比

6．下面的故事与（　）算法有着异曲同工之妙。 从前有座山，山里有座庙，庙里有个老和尚在给小和尚讲故事：“从前有座 山，山里有座庙，庙里有个老和尚在给小和尚讲故事：‘从前有座山，山里有座庙，庙里有个老和尚给小和尚讲故事……’”

A．枚举　　B．递归　　C．贪心　　D．分治

7．对于有n个顶点、m条边的无向联通图（m > n），需要删掉（　）条边才能使其成为一棵树。

A．n - 1　　B．m - n　　C．m - n - 1　　D．m - n + 1

8．5 个小朋友并排站成一列，其中有两个小朋友是双胞胎，如果要求这两个双胞胎必须相邻，则有（　）种不同排列方法?

A．48　　B．36　　C．24　　D．72

9．如果一棵二叉树只有根结点，那么这棵二叉树高度为1。请问高度为5的完全二叉树有（ ）种不同形态?

A．16　　B．15　　C．17　　D．32

10．有五副不同颜色的手套（共 10 只手套，每副手套左右手各 1 只），一次性从中取 6 只手套，请问恰好能配成两副手套的不同取法有（ ）种。

A．120　　B．180　　C．150　　D．30

11．一副纸牌除掉大小王有52张牌，四种花色，每种花色13张。假设从这52张牌中随机抽取13张纸牌，则至少（ ）张牌的花色一致。

A．4　　B．2　　C．3　　D．5

12．设 A 是一个实数的数组，考虑下面的递归算法：

```
XYZ (A[1..n])
if n= 1 then return A[1]
else temp ← XYZ (A[1..n-1])
    if temp < A[n]
    then return temp
    else return A[n]
```

请问算法XYZ的输出是什么？（ ）。

A．A数组的平均　　B．A数组的最小值　　C．A数组的中值　　D．A数组的最大值

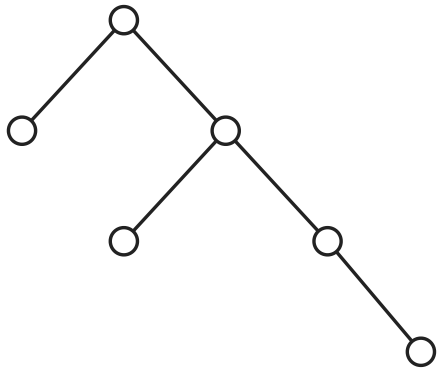
13．假设一棵二叉树的后序遍历序列为DGJHEBIFCA,中序遍历序列为DBGEHJACIF,则其前序遍历序列为（ ）。

A．ABCDEFGHIJ　　B．ABDEGHJCFI　　C．ABDEGJHCFI　　D．ABDEGHJFIC

14．一些数字可以颠倒过来看，例如0、1、8颠倒过来还是本身，6颠倒过来是9，9颠倒过来看还是6,其他数字颠倒过来都不构成数字。 类似的，一些多位数也可以颠倒过来看，比如106颠倒过来是901。假设某个城市的车牌只由5位数字组成，每一位都可以取0到9。 请问这个城市最多有多少个车牌倒过来恰好还是原来的车牌？（ ）

A．60　　B．125　　C．75　　D．100

15. 一棵二叉树如右图所示，若采用顺序存储结构，即用一维数组元素存储该二叉树中的结点（根结点的下标为1,若某结点的下标为i，则其左孩子位于下标2i处、右孩子位于下标 $2i+1$ 处），则该数组的最大下标至少为（　）。



A. 6　　B. 10　　C. 15　　D. 12

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 √，错误填 ×；共 18 小题，共计 36 分，各小题分值见题目）

(1)

```
#include <cstdlib>
#include <iostream>
using namespace std;

char encoder[26] = {'C','S','P',0};
char decoder[26];

string st;

int main()  {
    int k = 0;
    for (int i = 0; i < 26; ++i)
        if (encoder[i] != 0) ++k;
    for (char x ='A'; x <= 'Z'; ++x) {
        bool flag = true;
        for (int i = 0; i < 26; ++i)
            if (encoder[i] ==x) {
                flag = false;
                break;
            }
        if (flag) {
```

```
22              encoder[k]= x;
23              ++k;
24          }
25      }
26      for (int i = 0; i < 26; ++i)
27          decoder[encoder[i]- 'A'] = i + 'A';
28      cin >> st;
29      for (int i = 0; i < st.length( ); ++i)
30          st[i] = decoder[st[i] -'A'];
31      cout << st;
32      return 0;
33  }
```

判断题

16. (1.5分) 输入的字符串应当只由大写字母组成，否则在访问数组时可能越界。（ ）

17. (1.5分) 若输入的字符串不是空串，则输入的字符串与输出的字符串一定不一样。（ ）

18. (1.5分) 将第 12 行的 i < 26 改为 i < 16，程序运行结果不会改变。（ ）

19. (1.5分) 将第 26 行的 i < 26 改为 i < 16，程序运行结果不会改变。（ ）

单选题

20. (3分) 若输出的字符串为 ABCABCABCA，则下列说法正确的是（ ）。

A. 输入的字符串中既有 S 又有 P　B. 输入的字符串中既有 S 又有 B
C. 输入的字符串中既有 A 又有 P　D. 输入的字符串中既有 A 又有 B

21. (3分) 若输出的字符串为 CSPCSPCSPCSP，则下列说法正确的是（ ）。

A. 输入的字符串中既有 P 又有 A　B. 输入的字符串中既有 S 又有 R
C. 输入的字符串中既有 A 又有 S　D. 输入的字符串中既无 S 又无 A

(2)

```
01 #include <iostream>
02 using namespace std;
03
04 int n;
05 int d[1000];
06
07 int main() {
08     cin >> n;
09     for (int i = 0; i < n; ++i)
10         cin >> d[i];
11     int ans = -1;
12     for (int i = 0; i < n; ++i)
13         for (int j = 0; j < n; ++j)
14             if (d[i] < d[j])
15                 ans = max(ans, d[i] + d[j] - (d[i] & d[j]));
16     cout << ans;
17     return 0;
18 }
```

假设输入的n和d[i]都是不超过10000的正整数，完成下面的判断题和单选题：

判断题

22. (1.5分) n必须小于1000，否则程序可能会发生运行错误。（ ）

23. (1.5分) 输出一定大于等于0。（ ）

24. (1.5分) 若将第13行的“j = 0”改为“j = i + 1”，程序输出可能会改变。（ ）

25. (1.5分) 将第14行的“d[i] < d[j]”改为“d[i] != d[j]”，程序输出不会改变。（ ）

单选题

26. (3分) 若输入n为100，且输出为127，则输入的d[i]中不可能有（ ）。

A. 127　　B. 126　　C. 128　　D. 125

27. (3分) 若输出的数大于0，则下面说法正确的是（ ）。

A. 若输出为偶数，则输入的d[i]中最多有两个偶数
B. 若输出为奇数，则输入的d[i]中至少有两个奇数
C. 若输出为偶数，则输入的d[i]中至少有两个偶数
D. 若输出为奇数，则输入的d[i]中最多有两个奇数

(3)

```
01 #include<iostream>
02 using namespace std;
03 int n,m,i,j,p,k;
04 int a[100],b[100];
05 int main()
06 {
07     cin>>n>>m;
08     a[0]=n;i=0;p=0;k=1;
09     do{
10         for(j=0;j<i;j++)
11             if(a[i]==a[j])
12             {
13                 p=1;k=j;break;
14             }
15         if(p) break;
16         b[i]=a[i]/m;
17         a[i+1]=a[i]%m*10;
18         i++;
19     }while(a[i]!=0);
20     cout<<b[0]<<".";
21     for(j=1;j<k;j++) cout<<b[j];
22     if(p) cout<<"(";
23     for(j=k;j<i;j++) cout<<b[j];
24     if(p) cout<<") ";
25     cout<<endl;
26     return 0;
27 }
```

判断题

28. (1.5分) 程序输入的n和m不能相等。（ ）

29. (1.5分) 程序输入的m不能等于0。（ ）

30. (1.5分) 第9~19行的do…while循环一共有2个出口。（ ）

31. (1.5分) 数组a和b中的数值都小于或等于n。（ ）

单选题

32. (3分) 若输入数据为“11 8”,则输出结果为（ ）

A. 0.375　　B. 0.(375)　　C. 1.375　　D. 1.(375)

33. (3分) 若输人数据为“5 13”，则输出结果为（ ）。

A. 0.386514　　B. 0.(386514)　　C. 0.384615　　D. 0.(384615)

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **10** 小题，共计 **29** 分，各小题分值见题目）

(1)

完善程序:

（郊游活动）有 n 名同学参加学校组织的郊游活动，已知学校给这 n 名同学的郊游总经费为 A 元，与此同时第 i 位同学自己携带了 M_i 元。为了方便郊游，活动地点提供 $B(\geq n)$ 钢自行车供租用，租用第 i 钢自行车的价格为 C_i 元，每个同学可以使用自己携带的钱或者学校的郊游经费。为了方便账务管理，每位同学只能为自己租用自行车，且不会借钱给他人，他们想知道最多有多少位同学能够租用到自行车。

本题采用二分法。对于区间 $[l,r]$ ，我们取中间点 `mid` 并判断租用到自行车的人数能否达到 `mid`。判断的过程是利用贪心算法实现的。

```
01 #include <iostream>
02 using namespace std;
03 #define MAXN 1000000
04 int n, B, A, M[MAXN], C[MAXN], l, r, ans, mid;
05 bool check(int nn) {
06     int count = 0, i, j;
07     i = __(1)__;
08     j = 1;
09     while (i <= n) {
10         if(__(2)__)
11             count += C[j] - M[i];
12         i++;
13         j++;
14     }
15     return __(3)__;
16 }
```

```
17 void sort(int a[], int l, int r) {
18     int i = l, j = r, x = a[(l + r) / 2], y;
19     while (i <= j) {
20         while (a[i] < x) i++;
21         while (a[j] > x) j--;
22         if (i <= j) {
23             y = a[i]; a[i] = a[j]; a[j] = y;
24             i++; j--;
25         }
26     }
27     if (i < r) sort(a, i, r);
28     if (l < j) sort(a, l, j);
29 }
30 int main() {
31     int i;
32     cin >> n >> B >> A;
33     for (i = 1; i <= n; i++)
34         cin >> M[i];
35     for (i = 1; i <= B; i++)
36         cin >> C[i];
37     sort(M, 1, n);
38     sort(C, 1, B);
39     l = 0;
40     r = n;
41     while (l <= r) {
42         mid = (l + r) / 2;
43         if(__(4)__){
44             ans = mid;
45             l = mid + 1;
46         }else
47             r = __(5)__;
48     }
49     cout << ans << endl;
50     return 0;
51 }
```

34. (3分) (1)处应填：_____

35. (3分) (2)处应填：_____

36. (3分) (3)处应填：_____

37. (2.5分) (4)处应填：_____

38. (2.5分) (5)处应填：_____

(2)

(马走日)回溯算法实际上一个类似枚举的搜索尝试过程，主要是在搜索尝试过程中寻找问题的解，当发现已不满足求解条件时，就“回溯”返回，尝试其他路径。回溯法是一种选优搜索法按选向前乘以达到目标。但当搜索到某一步时，若发现原先选择并优或达不到目标，则返回一步重新选择，这种是不通就返回去再现的技术称为回溯法，而满足回溯条件的某个状态的点称为回溯点。

马在中国棋棋中以日字形规则移动。请编写一段程序，给定n×大小的棋盘以及马的初始位置(m,n)，要求不能重复盘上的同一个点计算马有多少途迹还可以逾历盘上的所有点。

```
01 #include<iostream>
02 using namespace std;
03 int r,c;
04 int cnt,tot;
05 int wayr[8]={2,2,1,-1,-2,-2,1,-1};
06 int wayc[8]={1,-1,2,2,1,-1,-2,-2};
07 bool mark[1001][1001];
08 bool check(int x, int y)
09 {
10     if(__(1)__) return true;
11     return false;
12 }
13 void search(int x,int y)
14 {
15     for(int i=0;i<8;i++)
16         if(__(2)__ && __(3)__)
17         {
18             mark[x+wayr[i]][y+wayc[i]]=true;
19             tot++;
20             if(__(4)__) cnt++;
21             search(__(5)__);
22             tot--;
23             mark[x+wayr[i]][y+wayc[i]]=false;
24         }
25 }
26 int main()
27 {
28     cnt=0;
29     int m,n;
```

```
30      cin>>r>>c>>m>>n;
31      if(!check(m,n)) cout<<0<<endl;
32      else if(r==1&&c==1) cout<<1<<endl;
33      else
34      {
35          mark[m][n]=true;
36          search(m,n);
37          cout<<cnt<<endl;
38      }
39  }
```

39. (3分) (1)处应填

A. x>=0 || y>=0 || x<y < c . B. $x \geqslant 0 \mid y \geqslant 0 \&\& x < r \mid y < e.$

C. x>=0 && y>=0 && x<r && y<c D. x>=0&&y>=0 || x<r &&y<c

40. (3分) (2)处应填

A. check(wayr[i], wayc[i]) B. check(x+wayr[i] , y+wayc[i])

C. !check(x+wayr[i],y+wayc[i]) D. !check(wayr[l],wayc[i])

41. (3分) (3)处应填

A. mark[wayr[i]][wayc[i]] B. mark[x+wayr[i]][y+wayc[ii]]

C. !mark[wayr[i]][wayc[i]) D. !mark[x+wayr[i]][y+wayc[i]]

42. (3分) (4)处应填

A. tot==r * c-1 B. tot-=r * c C. cnt==r * c-1 D. cnt==r * c

43. (3分) (5)处应填

A. wayr[i], wayc[i] B. x-wayr[i], y-wayc[i]

C. x-wayr[i], y+wayc[i] D. x+wayr[i], y+wayc[i]

CSP-J 初赛模拟卷（第 11 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

1．1TB代表的字节数是（ ）。

A．2的10次方　　B．2的20次方　　C．2的30次方　　D．2的40次方

2．在 PC 机中，PENTIUM(奔腾)、酷睿、赛扬等是指（ ）。

A．生产厂家名称　　B．硬盘的型号　　C．CPU 的型号　　D．显示器的型号

3．中国计算机学会于（ ）年创办全国青少年计算机程序设计竞赛。

A．1983　　B．1984　　C．1985　　D．1986

4．（ ）不属于操作系统。

A．Windows　　B．DOS　　C．Photoshop　　D．NOI Linux

5．FTP 可以用于（ ）。

A．远程传输文件　　B．发送电子邮件　　C．浏览网页　　D．网上聊天

6．以下不属于无线通信技术的是（ ）。

A．蓝牙　　B．Wifi　　C．GPRS　　D．以太网

7．以下哪个奖项是计算机科学领域的最高奖？（ ）

A．图灵奖　　B．鲁班奖　　C．诺贝尔奖　　D．普利策奖

8．二进制数101.11对应的十进制数是（ ）。

A．6.5　　B．5.5　　C．5.75　　D．5.25

9．10 个三好学生名额分配到 7 个班级，每个班级至少有一个名额，一共有（ ）种不同的分配方案。

A．84　　B．72　　C．56　　D．504

10．下列各无符号十进制整数中，能用八位二进制表示的数中最大的是（ ）。

A．296　　B．133　　C．256　　D．199

11．甲、乙、丙三位同学选修课程，从 4 门课程中，甲选修 2 门，乙、丙各选修3门，则不同的选修方案共有（ ）种。

A．36　　B．48　　C．96　　D．192

12．关于拓扑排序，下面说法正确的是（ ）。

A．所有连通的有向图都可以实现拓扑排序

B．对同一个图而言，拓扑排序的结果是唯一的

C．拓扑排序中入度为0的结点总会排在入度大于0的结点的前面

D．拓扑排序结果序列中的第一个结点一定是入度为0的点

13．仿生学的问世开辟了独特的科学技术发展道路。人们研究生物体的结构、功能和工作原理，并将这些原理移植于新兴的工程技术中。以下关于仿生学的叙述，错误的是（ ）。

A．由研究蝙蝠，发明雷达　B．由研究蜘蛛网，发明因特网

C．由研究海豚，发明声呐　D．由研究电鱼，发明伏特电池

14．甲乙丙丁四人在考虑周末要不要外出郊游。 已知①如果周末下雨，并且乙不去，则甲一定不去；②如果乙去，则丁一定去；③如果丙去，则丁一定不去；④如果丁不去，而且甲不去，则丙一定不去。 如果周末丙去了，则甲_____，乙_____，丁_____，周末_____。

（1）甲：

A．去了；　B．没去。

（2）乙：

A．去了；　B．没去。

（3）丁：

A．去了；　B．没去。

（4）周末：

A．下雨；　B．没下雨。

15．（ ）是一种选优搜索法，按选优条件向前搜索，以达到目标。当搜索到某一步时，发现原先选择并不优或达不到目标，就退回一步重新选择。

A．回溯法　　B．枚举法　　C．动态规划　　D．贪心

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 **√**，错误填 **×**；共 **18** 小题，共计 **35.5** 分，各小题分值见题目）

(1)

```
01 #include <stdio.h>
02
03 int n;
04 int a[1000];
05
06 int f(int x)
07 {
08     int ret = 0;
09     for (; x; x &= x - 1) ret++;
10     return ret;
11 }
12
13 int g(int x)
14 {
15     return x & -x;
16 }
17
18 int main()
19 {
20     scanf("%d", &n);
21     for (int i = 0; i < n; i++) scanf("%d", &a[i]);
22     for (int i = 0; i < n; i++)
23         printf("%d ", f(a[i]) + g(a[i]));
24     printf("\n");
25     return 0;
26 }
27
```

判断题

16. (1.5分) 输入的 n 等于1001时，程序不会发生下标越界。（ ）

17. (1.5分) 输入的 $a[i]$ 必须全为正整数，否则程序将陷入死循环。（ ）

18. (1.5分) 当输入为“5 2 11 9 16 10”时，输出为“3 4 3 17 5”。（ ）

19. (1.5分) 当输入为”1 511998”时，输出为”18”。（ ）

20. (1.5分) 将源代码中 g 函数的定义(13-16行)移到main函数的后面，程序可以正常编译运行。（ ）

单选题

21. (3分) 当输入为”2 -65536 2147483647”时，输出为（ ）

A. “65532 33”　　B. “65552 32”　　C. “65535 34”　　D. “65554 33”

(2)

```
01 #include<cstdio>
02 using namespace std;
03 int n, m;
04 int a[100], b[100];
05
06 int main() {
07     scanf("%d%d", &n, &m);
08     for (int i = 1; i <= n; ++i)
09         a[i] = b[i] = 0;
10     for (int i = 1; i <= m; ++i) {
11         int x, y;
12         scanf("%d%d", &x, &y);
13         if (a[x] < y && b[y] < x) {
14             if (a[x] > 0)
15                 b[a[x]] = 0;
16             if (b[y] > 0)
17                 a[b[y]] = 0;
18             a[x] = y;
19             b[y] = x;
20         }
21     }
22     int ans = 0;
23     for (int i = 1; i <= n; ++i) {
24         if (a[i] == 0)
25             ++ans;
26         if (b[i] == 0)
27             ++ans;
28     }
29     printf("%d", ans);
30     return 0;
31 }
```

假设输入的 n 和 m 都是正整数，x 和 y 都是在 $[1,n]$ 的范围内的整数，完成下面的判断题和单选题：

判断题

22. (1.5分) 当 $m>0$ 时，输出的值一定小于 $2n$。（ ）

23. (1.5分) 执行完第27行的“+ + ans”时，ans一定是偶数。（ ）

24. (1.5分) $a[i]$ 和 $b[i]$ 不可能同时大于0。（ ）

25. (1.5分) 右程序执行到第13行时， x 总是小于 y ，那么第15行不会被执行。（ ）

单选题

26. (3分) 若 m 个 x 两两不同，且 m 个 y 两两不同，则输出的值为（ ）

A. $2n-2m$　　B. $2n+2$　　C. $2n-2$　　D. $2n$

27. (3分) 若 m 个 x 两两不同，且 m 个 y 都相等，则输出的值为（ ）

A. $2n-2$　　B. $2n$　　C. $2m$　　D. $2n-2m$

(3)

```
01 #include<cstdio>
02 #include<cstring>
03 using namespace std;
04 int n, m;
05 int f[101][101];
06 int F[101][101];
07 int main(){
08     scanf("%d%d",&n,&m);//n的值在1到100之间
09     memset(f,-1,sizeof(f));
10     for(int i = 1; i <=  m; i++) {
11         int u,v,w;//w的值在0到10000之间
12         scanf("%d%d%d",&u,&v,&w);
13         f[u][v]=f[v][u]=w;
14     }
15     for(int k = 1; k <= n; k++)
16         for(int i=1;i <=n; i++)
17             for(int j=1;j<= n; j++)
18                 if(f[i][k]!=-1&&f[k][j]!=-1)
19                     if(f[i][j]==-1||f[i][j]>f[k][j]+f[i][k])
20                         f[i][j]=f[i][k]+f[k][j];
```

```
21      int ans=2147483647;
22      for(int i=1;i<=n;i++)
23          for(int j=1;j<=n; j++){
24              for(int x=1;x<=n; x++)
25                  for(int y=1;y<=n; y++)
26                      F[x][y]=f[x][y];
27              F[i][j]=F[j][i]=0;
28              for(int x=1;x<= n; x++)
29                  for(int y=1;y<=n; y++)
30                      if(F[x][y]==-1||F[x][y]>F[x][i]+F[i][y])
31                          F[x][y]=F[x][i]+F[i][y];
32              for(int x=1;x<=n; x++)
33                  for(int y=1;y<=n; y++)
34                      if(F[x][y]==-1||F[x][y]>F[x][j]+F[j][y])
35                          F[x][y]=F[x][j]+F[j][y];
36              int res=0;
37              for(int x=1;x<=n; x++)
38                  for(int y=1;y<x; y++)
39                      res +=F[x][y];
40              ans = res < ans ? res : ans;
41          }
42      printf("%d\n",ans);
43      return 0;
44  }
```

判断题

28. (1分) 14到16行，将外层到内层的循环变量依次调整为i、j、k，程序的运行的结果不变。（ ）

29. (2分) 这个程序的时间复杂度和m无关。（ ）

30. (2分) 20行的ans如果初始化为10^7时，可能无法得到正确结果。（ ）

31. (2分) 若将第27到30行的部分和31到34行的两个部分互换，程序的运行的结果不变。（ ）

单选题

32. (3分) 若输入数据为4 5/1 2 3/1 3 6/2 3 4/2 4 7/3 4 2(其中“/”为换行符)，则输出为（ ）。

A. 14　　B. 18　　C. 21　　D. 28

33. (3分) 这个程序使用了（ ）算法。

A. Floyd　　B. Dijkstra　　C. Prim　　D. Kruskal

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **10** 小题，共计 **28** 分，各小题分值见题目）

(1)

完善程序:

(中位数 median) 给定 n(n 为奇数且小于 1000)个整数，整数的范围在 $0-m(0<m<2^{31})$ 之间，请使用二分法求这 n 个整数的中位数。所谓中位数，是指将这 n 个数排序之后，排在正中间的数。

```
01 #include <iostream>
02 using namespace std;
03 const int MAXN = 1000;
04 int n, i, lbound, rbound, mid, m, count;
05 int x[MAXN];
06 int main()
07 {
08     cin >> n >> m;
09     for (i = 0; i < n; i++)
10         cin >> x[i];
11     lbound = 0;
12     rbound = m;
13     while (__(1)__)
14     {
15         mid = lbound + (rbound - lbound) / 2;
16         __(2)__;
17         for (i = 0; i < n; i++)
18             if (__(3)__)
19                 __(4)__;
20         if (count > n / 2)
21             lbound = mid + 1;
22         else
23             __(5)__;
24         cout << mid << " " << lbound << " " << rbound << " " <<
   count << endl;
25     }
26     cout << rbound << endl;
27     return (0);
28 }
```

34. (3分) (1)处应填：_____

35. (3分) (2)处应填：_____

36. (3分) (3)处应填：_____

37. (3分) (4)处应填：_____

38. (2分) (5)处应填：_____

(2)

完善程序:

（二叉查找树）二叉查找树具有如下性质:每个节点的值都大于其左子树上所有节点的值、小于其右子树上所有节点的值。试判断一棵树是否为二叉查找树。输入的第一个行包含一个整数 n，表示这棵树有 n 个顶点，编号分别为 $1,2,...,n$，其中编号为1的为根结点。之后的第1行有三个数value, left_child, right_child，分别表示该节点关键字的值、左子节点的编号、右子节点的编号；如果不存在左子节点或右子节点，则用0代替。输出1表示这棵树是二叉查找树，输出0则表示不是。

```
01 #include <iostream>
02 using namespace std;
03 const int SIZE = 100;
04 const int INFINITE = 1000000;
05 struct node
06 {
07     int left_child, right_child, value;
08 }; node a[SIZE];
09 int is_bst( int root, int lower_bound, int upper_bound )
10 {
11     int cur;
12     if ( root == 0 )
13         return(1);
14     cur = a[root].value;
15     if ( (cur > lower_bound) && ( __(1)__ ) && (is_bst(
   a[root].left_child, lower_bound, cur ) == 1) && (is_bst( __(2)__,
   __(3)__, __(4)__ ) == 1) )
16         return(1);
17     return(0);
18 }
19 int main()
20 {
```

```
21      int i, n; cin >> n;
22      for ( i = 1; i <= n; i++ )
23          cin >> a[i].value >> a[i].left_child >> a[i].right_child;
24      cout << is_bst( __(5)__, -INFINITE, INFINITE ) << endl;
25      return(0);
26  }
```

39. (3分) (1)处应填：_____

40. (3分) (2)处应填：_____

41. (3分) (3)处应填：_____

42. (3分) (4)处应填：_____

43. (2分) (5)处应填：_____

CSP-J 初赛模拟卷（第 12 套）

入门级 C++ 语言试题

一、单项选择题（共 **14** 题，每题 **2** 分，共计 **28** 分）

1．操作系统的作用是（ ）。

A．把源程序译成目标程序　B．便于进行数据管理

C．控制和管理系统资源　D．实现硬件之间的连接

2．NOI 的中文意思是（ ）。

A．中国信息学联赛　B．全国青少年信息学奥林匹克竞赛

C．中国青少年信息学奥林匹克竞赛　D．中国计算机协会

3．（ ）是一种先进先出的线性表。

A．栈　B．队列　C．哈希表（散列表）　D．二叉树

4．如果256 种颜色用二进制编码来表示，至少需要（ ）位。

A．6　B．7　C．8　D．9

5．设 $x = true, y = true, z = false$，以下逻辑运算表达式值为真的是（ ）。

A．$(y \vee z) \wedge x \wedge z$　B．$x \wedge (z \vee y) \wedge z$　C．$(x \wedge y) \wedge z$　D．$(x \wedge y) \vee (z \vee x)$

6．下列四个不同进制的数中，与其它三项数值上不相等的是

A．$(269)_{16}$　B．$(617)_{10}$　C．$(1151)_{8}$　D．$(1001101011)_{2}$

7．表达式 $a*(b+c)*d$ 的后缀表达式为（ ），其中“*”和“+”是运算符。

A．**a+bcd　B．abc+*d*　C．abc+d**　D．*a*+bcd

8．一副纸牌除掉大小王有52张牌，四种花色，每种花色13张。假设从这52张牌中随机抽取13张纸牌，则至少（ ）张牌的花色一致。

A．4　B．2　C．3　D．5

9．前序遍历序列与中序遍历序列相同的二叉树为（ ）。

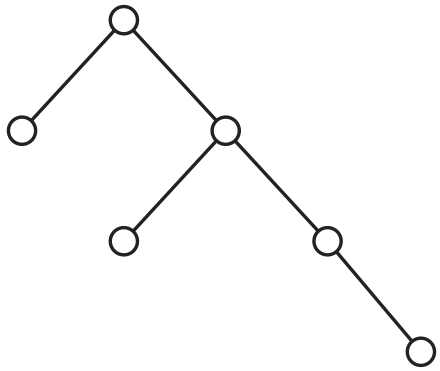
A．根结点无左子树

B．根结点无右子树

C．只有根结点的二叉树或非叶子结点只有左子树的二叉树

D．只有根结点的二叉树或非叶子结点只有右子树的二叉树

10. 一棵二叉树如右图所示，若采用顺序存储结构，即用一维数组元素存储该二叉树中的结点（根结点的下标为1，若某结点的下标为 i ，则其左孩子位于下标2i处、右孩子位于下标 $(2i+1)$ 处），则图中所有结点的最大下标为（ ）。



A. 6　　B. 10　　C. 12　　D. 15

11. 原字符串中任意一段连续的字符所组成的新字符串称为子串。 则字符“AAABBBCCC”共有（ ）个不同的非空子串。

A. 3　　B. 12　　C. 36　　D. 45

12. 现有一段文言文，要通过二进制哈夫曼编码进行压缩。简单起见，假设这段文言文只由4个汉字“之”、“乎”、“者”、“也”组成，它们出现的次数分别为700、600、300、400。若待合并结点权值相同，优先选取由其他结点合并形成的结点。那么，“也”字的编码长度可能是（ ）。

A. 1　　B. 2　　C. 3　　D. 4

13. 设 G 是有 n 个结点、m 条边 ($n \leq m$) 的连通图，必须删去 G 的（ ）条边，才能使得 G 变成一棵树。

A. $m-n+1$　　B. m - n　　C. $m+n+1$　　D. $n-m+1$

14. 双向链表中有两个指针域llink和rlink，分别指向该结点的前驱及后继。设p指向链表中的一个结点，它的左右结点均非空。现要求删除结点p，则下面语句序列中错误的是（ ）。

A. p->rlink->llink = p->rlink; p->llink->rlink = p->llink; delete p;
B. p->llink->rlink = p->rlink; p->rlink->llink = p->llink; delete p;
C. p->rlink->llink = p->llink; p->rlink->llink->rlink = p->rlink; delete p;
D. p->llink->rlink = p->rlink; p->llink->rlink->llink = p->llink; delete p;

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 √，错误填 ×；共 18 小题，共计 36 分，各小题分值见题目）

(1)

```
01 #include<iostream>
02 using namespace std;
03 int main()
04 {
05     const int SIZE=10;
06     int height[SIZE],num[SIZE],n,ans;
07     cin>>n;
08     for(int i=0;i<n;i++){
09         cin>>height[i];
10         num[i]=1;
11         for(int j=0;j<i;j++){
12             if((height[j]<height[i])&&(num[j]>=num[i]))
13                 num[i]=num[j]+1;
14         }
15     }
16     ans=0;
17     for(int i=0;i<n;i++){
18         if(num[i]>ans) ans=num[i];
19     }
20     cout<<ans<<endl;
21     return 0;
22 }
```

判断题

15. (1.5分) 如果height数组中的输入有负数，则程序会出错。（ ）

16. (1.5分) 程序输出的ans小于或等于n。（ ）

17. (1.5分) 将12行num[j]>=num[i]“改为”num[j]>num[i]“，程序的输出结果不会改变。（ ）

18. (1.5分) 将18行”num[i]>ans”改为”num[i]>=ans”，程序的输出结果不会改变。（ ）

单选题

19. (3分) 若输入的数据为 10 1 1 1 1 1 1 1 1 1 则程序的输出结果是（ ）。

A. 4　　B. 3　　C. 2　　D. 1

20. (3分) 若输入的数据为 10 3 2 5 11 12 7 4 10 15 6 因程序的输出结果是（ ）。

A. 4　　B. 5　　C. 6　　D. 7

(2)

```
01 #include <iostream>
02 using namespace std;
03
04 int n;
05 int d[1000];
06
07 int main() {
08     cin >> n;
09     for (int i = 0; i < n; ++i)
10         cin >> d[i];
11     int ans = -1;
12     for (int i = 0; i < n; ++i)
13         for (int j = 0; j < n; ++j)
14             if (d[i] < d[j])
15                 ans = max(ans, d[i] + d[j] - (d[i] & d[j]));
16     cout << ans;
17     return 0;
18 }
```

假设输入的n和d[i]都是不超过10000的正整数，完成下面的判断题和单选题：

判断题

21. (1.5分) n必须小于1000，否则程序可能会发生运行错误。（ ）

22. (1.5分) 输出一定大于等于0。（ ）

23. (1.5分) 若将第13行的“j = 0”改为“j = i + 1”，程序输出可能会改变。（ ）

24. (1.5分) 将第14行的“d[i] < d[j]”改为“d[i] != d[j]”，程序输出不会改变。（ ）

单选题

25. (3分) 若输入n为100，且输出为127，则输入的d[i]中不可能有（ ）。

A. 127　　B. 126　　C. 128　　D. 125

26. (3分) 若输出的数大于0，则下面说法正确的是（ ）。

A. 若输出为偶数，则输入的d[i]中最多有两个偶数
B. 若输出为奇数，则输入的d[i]中至少有两个奇数
C. 若输出为偶数，则输入的d[i]中至少有两个偶数
D. 若输出为奇数，则输入的d[i]中最多有两个奇数

(3)

```
01 #include<cstdio>
02 using namespace std;
03 int n, m;
04 int a[100], b[100];
05
06 int main() {
07     scanf("%d%d", &n, &m);
08     for (int i = 1; i <= n; ++i)
09         a[i] = b[i] = 0;
10     for (int i = 1; i <= m; ++i) {
11         int x, y;
12         scanf("%d%d", &x, &y);
13         if (a[x] < y && b[y] < x) {
14             if (a[x] > 0)
15                 b[a[x]] = 0;
16             if (b[y] > 0)
17                 a[b[y]] = 0;
18             a[x] = y;
19             b[y] = x;
20         }
21     }
22     int ans = 0;
```

```
23      for (int i = 1; i <= n; ++i) {
24          if (a[i] == 0)
25              ++ans;
26          if (b[i] == 0)
27              ++ans;
28      }
29      printf("%d", ans);
30      return 0;
31  }
```

假设输入的 n 和 m 都是正整数，x 和 y 都是在 $[1,n]$ 的范围内的整数，完成下面的判断题和单选题:· 判断题

判断题

27. (1.5分) 当m>0时，输出的值一定小于2n。（ ）

28. (1.5分) 执行完第27行的“++ans”时，ans一定是偶数。（ ）

29. (1.5分) $a[i]$ 和 $b[i]$ 不可能同时大于0。（ ）

30. (1.5分) 若每次执行到第13行时都有 x < y，则第15行不会被执行。（ ）

单选题

31. (3分) 若 m 个 x 两两不同，且 m 个 y 两两不同，则输出的值为（ ）

A. $2n-2m$　　B. $2n+2$　　C. $2n-2$　　D. $2n$

32. (3分) 若 m 个 x 两两不同，且 m 个 y 都相等，则输出的值为（ ）

A. $2n-2$　　B. $2n$　　C. $2m$　　D. $2n-2m$

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **15** 小题，共计 **43** 分，各小题分值见题目）

(1)

（质因数分解）给出正整数 n ，请输入将 n 质因数分解的结果，结果从小到大输出。例如:输入 $n=120$ ，程序应该输出22235，表示: $120=2\times2\ \ 2\times3\times5$ ，输入保证

$2 \leq n \leq 10^0$ 。提示:先从小到大枚举变量 i ，然后用 i 不停试除 n 来寻找所有的质因子。试补全程序。

```
#include <cstdio>
using namespace std;
int n, i;

int main() {
    scanf("%d", &n);
    for(i = __(1)__; __(2)__ <=n; i ++){
        __(3)__{
            printf("%d ", i);
            n = n / i;
        }
    }
    if(__(4)__)
        printf("%d ", __(5)__);
    return 0;
}
```

33. (3分) (1)处应填

A. 1　　B. n-1　　C. 2　　D. 0

34. (3分) (2)处应填

A. n/i　　B. n/(i*i)　　C. i*i　　D. i*i*i

35. (3分) (3)处应填

A. if(n%i==0)　B. if(i*i<=n)　C. while(n%i==0)　D. while(i*i<=n)

36. (3分) (4)处应填

A. n>1　　B. n<=1　　C. i<n/i　　D. i+i<=n

37. (3分) (5)处应填

A. 2　　B. n/i　　C. n　　D. i

(2)

完善程序

（最大公约数之和）下列程序想要求解整数 n 的所有约数两两之间最大公约数的和对10007求余后的值，试补全程序。举例来说，4的所有约数是1,2,4。1和2的最大公约数为1；2和4

的最大公约数为2；1和4的最大公约数为1。于是答案为 $1+2+1=4$ 。要求getDivisor函数的复杂度为 $O(\sqrt{n})$，gcd函数的复杂度为 $O(\log \max(a,b))$ 。

```
01 #include <iostream>
02 using namespace std;
03 const int N = 110000, P = 10007;
04 int n;
05 int a[N], len;
06 int ans;
07 void getDivisor() {
08     len = 0;
09     for (int i = 1; __(1)__ <= n; ++i)
10         if (n % i == 0) {
11             a[++len] = i;
12             if ( __(2)__ != i) a[++len] = n / i;
13         }
14 }
15 int gcd(int a, int b) {
16     if (b == 0) {
17         __(3)__ ;
18     }
19     return gcd(b, __(4)__ );
20 }
21 int main() {
22     cin >> n;
23     getDivisor();
24     ans = 0;
25     for (int i = 1; i <= len; ++i) {
26         for (int j = i + 1; j <= len; ++j) {
27             ans = ( __(5)__ ) % P;
28         }
29     }
30     cout << ans << endl;
31     return 0;
32 }
```

38. (2分) (1)处应填：_____

39. (3分) (2)处应填：_____

40. (3分) (3)处应填：_____

41. (3分) (4)处应填：_____

42. (3分) (5)处应填：_____

(3)

完善程序:

（切割绳子）有 n 条绳子，每条绳子的长度已知且均为正整数。绳子可以以任意正整数长度切割，但不可以连接。现在要从这些绳子中切割出 m 条长度相同的绳段，求绳段的最大长度是多少。

输入:第一行是一个不超过100的正整数 n，第二行是 n 个不超过 10^6 的正整数，表示每条绳子的长度，第三行是一个不超过 10^8 的正整数 m。输出:绳段的最大长度，若无法切割，输出Failed。

```
01 #include<iostream>
02 using namespace std;
03 int n, m, i, lbound, ubound, mid, count;
04 int len[100]; // 绳子长度
05 int main()
06 {
07     cin >> n;
08     count = 0;
09     for (i = 0; i < n; i++)
10     {
11         cin >> len[i];
12         __(1)__;
13     }
14     cin >> m;
15     if (__(2)__)
16     {
17         cout << "Failed" << endl;
18         return 0;
19     }
20     lbound = 1;
21     ubound = 1000000;
22     while (__(3)__)
23     {
24         mid = __(4)__;
25         count = 0;
26         for (i = 0; i < n; i++)
27             __(5)__;
28         if (count < m)
29             ubound = mid - 1;
```

```
30          else
31              lbound = mid;
32      }
33      cout << lbound << endl;
34      return 0;
35  }
```

43. (2.5分) (1)处应填：_____

44. (2.5分) (2)处应填：_____

45. (3分) (3)处应填：_____

46. (3分) (4)处应填：_____

47. (3分) (5)处应填：_____

CSP-J 初赛模拟卷（第 13 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

1．生物特征识别，是利用人体本身的生物特征进行身份认证的一种技术。目前，指纹识别、虹膜识别、人脸识别等技术已广泛应用于政府、银行、安全防卫等领域。以下不属于生物特征识别技术及其应用的是（ ）。

A．指静脉验证　　B．步态验证　　C．ATM机密码验证　　D．声音验证

2．一个正整数在二进制下有100位，则它在十六进制下有（ ）位。

A．7　　B．13　　C．25　　D．不能确定

3．以下哪个是面向对象的高级语言（ ）。

A．汇编语言　　B．C++　　C．Fortran　　D．Basic

4．Wi-Fi 主要用于构建哪类无线网络？（ ）

A．无线广域网　　B．无线城域网　　C．无线局域网　　D．无线路由器

5．以下逻辑表达式的值恒为真的是（ ）。

A．$P \vee (\neg P \wedge Q) \vee (\neg P \wedge \neg Q)$　　B．$Q \vee (\neg P \wedge Q) \vee (P \wedge \neg Q)$

C．$P \vee Q \vee (P \wedge \neg Q) \vee (\neg P \wedge Q)$　　D．$P \vee \neg Q \vee (P \wedge \neg Q) \vee (\neg P \wedge \neg Q)$

6．把 64 位非零浮点数强制转换成32 位浮点数后，不可能（ ）。

A．大于原数　　B．小于原数　　C．等于原数　　D．与原数符号相反

7．前缀表达式 + 3 * 2 + 5 12的值是（ ）。

A．23　　B．25　　C．37　　D．65

8．计算机界的最高奖是（ ）。

A．菲尔兹奖　　B．诺贝尔奖　　C．图灵奖　　D．普利策奖

9．甲、乙、丙三位同学选修课程，从 4 门课程中，甲选修 2 门，乙、丙各选修3门，则不同的选修方案共有（ ）种。

A．36　　B．48　　C．96　　D．192

10. 如果一棵二叉树的中序遍历是 BAC，那么它的先序遍历不可能是（ ）。

A. ABC　　B. CBA　　C. ACB　　D. BAC

11. 将（2,6,10,17）分别存储到某个地址区间为0~10的哈希表中，如果哈希函数 $h(x) = (\ )$，将不会产生冲突，其中 $a \mod b$ 表示 a 除以 b 的余数。

A. $x \mod 11$　　B. $x^2 \mod 11$

C. $(2x) \mod 11$　　D. $\lfloor sqrt(x) \rfloor \mod 11$，其中 $\lfloor sqrt(x) \rfloor$ 表示 $\sqrt{x}$ 下取整

12. 向一个栈顶指针为 hs 的链式栈中插入一个指针 s 指向的结点时，应执行（ ）。

A. hs->next = s;　　B. s->next = hs; hs = s;

C. s->next = hs->next; hs->next = s;　　D. s->next = hs; hs = hs->next;

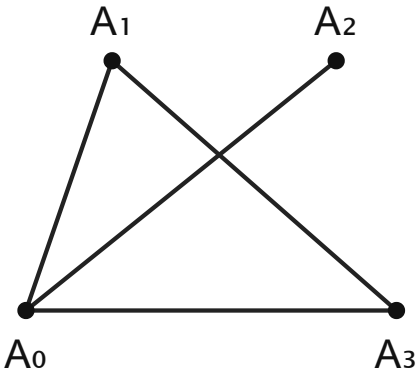
13. 使用冒泡排序对序列进行升序排列，每执行一次交换操作系统将会减少 1 个逆序对，因此序列 5，4，3，2，1 需要执行（ ）次操作，才能完成冒泡排序。

A. 0　　B. 5　　C. 10　　D. 15

14. 设 A 和 B 是两个长为 n 的有序数组，现在需要将 A 和 B 合并成一个排好序的数组，任何以元素比较作为基本运算的归并算法在最坏情况下至少要做（ ）次比较。

A. n^2　　B. nlogn　　C. 2n　　D. 2n - 1

15. 以 A_0 作为起点，对下面的无向图进行深度优先遍历时，遍历顺序不可能是（ ）。



A. A_0, A_1, A_2, A_3　　B. A_0, A_1, A_3, A_2　　C. A_0, A_2, A_1, A_3　　D. A_0, A_3, A_1, A_2

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 **√**，错误填 **×**；共 **18** 小题，共计 **37** 分，各小题分值见题目）

(1)

```
01 #include <cstdio>
02 #include <cstring>
03 using namespace std;
04 char st[100001];
05 int main() {
06     scanf("%s", st);
07     int n = strlen(st);
08     for (int i = 1; i <= n; ++i) {
09         if (n % i == 0) {
10             char c = st[i - 1];
11             if (c >= 'a')
12                 st[i - 1] = c - 'a' + 'A';
13         }
14     }
15     printf("%s", st);
16     return 0;
17 }
```

判断题

16. (1.5分) 输入的字符串只能由小写字母或大写字母组成。（ ）

17. (1.5分) 若将第 8 行的 `i = 1` 改为 `i = 0`，程序运行时会发生错误。（ ）

18. (1.5分) 若将第 8 行的 `i <= n` 改为 `i * i <= n`，程序运行结果不会改变。（ ）

19. (1.5分) 若输入的字符串全部由大写字母组成，那么输出的字符串就跟输入的字符串一样。（ ）

单选题

20. (3分) 若输入的字符串长度为 18，那么输入的字符串跟输出的字符串相比，至多有（ ）个字符不同。

A. 18　　B. 6　　C. 10　　D. 1

21. (3分) 若输入的字符串长度为（ ），那么输入的字符串跟输出的字符串相比，至多有 36 个字符不同。

A. 36　　B. 100000　　C. 1　　D. 128

(2)

```
01 #include<iostream>
02 using namespace std;
03 int n,m,i,j,p,k;
04 int a[100],b[100];
05 int main()
06 {
07     cin>>n>>m;
08     a[0]=n;i=0;p=0;k=1;
09     do{
10         for(j=0;j<i;j++)
11             if(a[i]==a[j])
12             {
13                 p=1;k=j;break;
14             }
15         if(p) break;
16         b[i]=a[i]/m;
17         a[i+1]=a[i]%m*10;
18         i++;
19     }while(a[i]!=0);
20     cout<<b[0]<<".";
21     for(j=1;j<k;j++) cout<<b[j];
22     if(p) cout<<"(";
23     for(j=k;j<i;j++) cout<<b[j];
24     if(p) cout<<") ";
25     cout<<endl;
26     return 0;
27 }
```

判断题

22. (1.5分) 程序输入的n和m不能相等。（ ）

23. (1.5分) 程序输入的m不能等于0。（ ）

24. (1.5分) 第9~19行的do…while循环一共有2个出口。（ ）

25. (1.5分) 数组a和b中的数值都小于或等于n。（ ）

单选题

26. (3分) 若输入数据为“11 8”，则输出结果为（ ）

A. 0.375　　B. 0.(375)　　C. 1.375　　D. 1.(375)

27. (3分) 若输人数据为“5 13”，则输出结果为（ ）。

A. 0.386514　　B. 0.(386514)　　C. 0.384615　　D. 0.(384615)

(3)

```
01 #include <iostream>
02 #include <cstdlib>
03 using namespace std;
04
05 int n;
06 int d[10000];
07
08 int find(int L, int R, int k) { if (L == R) return d[L];
09     int x = rand() % (R - L + 1) + L;
10     swap(d[L], d[x]);
11     int a = L + 1, b = R;
12     while (a < b) {
13         while (a < b && d[a] < d[L])
14             ++a;
15         while (a < b && d[b] >= d[L])
16             --b;
17         swap(d[a], d[b]);
18     }
19     if (d[a] < d[L])
20         ++a;
21     if (a - L == k)
22         return d[L];
23     if (a - L < k)
24         return find(a, R, k - (a - L));
25     return find(L + 1, a - 1, k);
26 }
27
28 int main() {
29     int k;
30     cin >> n;
31     cin >> k;
32     for (int i = 0; i < n; ++i)
```

```
33          cin >> d[i];
34      cout << find(0, n - 1, k);
35      return 0;
36  }
```

假设 $1 \le k \le n \le 10000$，输入数组元素为不超过 10000 的正整数，且 rand() 产生均匀随机数。复杂度选项均指渐进数量级。

判断题

28. (1.5分) 第 9 行中 x 的数值范围为 $[L+1, R]$。（ ）

29. (1.5分) 将第 19 行中的 d[a] 改为 d[b]，程序不会发生运行错误。（ ）

单选题

30. (2.5分) 当输入数组严格单调递增时，此程序的平均时间复杂度为（ ）。

A. $O(n \log n)$　　B. $O(n)$　　C. $O(\log n)$　　D. $O(n^2)$

31. (2.5分) 当输入数组严格单调递减时，此程序的平均时间复杂度为（ ）。

A. $O(n^2)$　　B. $O(n)$　　C. $O(n \log n)$　　D. $O(\log n)$

32. (2.5分) 当输入数组严格单调递增时，此程序①平均时间复杂度与②最坏时间复杂度分别为（ ）。

A. $O(n)$，$O(n^2)$　　B. $O(n)$，$O(n \log n)$
C. $O(n \log n)$，$O(n^2)$　　D. $O(n \log n)$，$O(n \log n)$

33. (2.5分) 若所有数组元素相等，且 $k = n$，此程序的时间复杂度为（ ）。

A. $O(n)$　　B. $O(\log n)$　　C. $O(n \log n)$　　D. $O(n^2)$

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **10** 小题，共计 **28** 分，各小题分值见题目）

(1)

```
01  const int SIZE = 100;
02  int a[SIZE], n;
```

它记录着一个长度为n的序列 $a[1], a[2], \cdots, a[n]$ 。 现在需要一个函数，以整数 $p(1 \le p \le n)$ 为参数，实现如下功能:将序列a的前p个数与后n-p个数对调，且不改变这p个数（或n-p个数）之间的相对位置。例如，长度为5的序列1,2,3,4,5，当p=2时重排结果为

3,4,5,1,2。 有一种朴素的算法可以实现这一需求，其时间复杂度为 $O(n)$ 、空间复杂度为 $O(n)$ ：

```
01 void swap1( int p )
02 {
03 int i, j, b[SIZE];
04 for ( i = 1; i <= p; i++ )
05 b[__(1)__] = a[i];
06 for ( i = p + 1; i <= n; i++ )
07 b[i - p] = __(2)__;
08 for ( i = 1; i <= __(3)__; i++ )
09 a[i] = b[i];
10 }
```

我们也可以用时间换空间，使用时间复杂度为 $O(n^2)$ 、空间复杂度为 $O(1)$ 的算法：

```
01 void swap2(int p)
02 {
03     int i, j, temp;
04     for ( i = p + 1; i <= n; i++ )
05     {
06         temp = a[i];
07         for ( j = i; j >= __(4)__; j-- )
08             a[j] = a[j - 1];
09         __(5)__ = temp;
10     }
11 }
```

34. (3分) (1)处应填：_____

35. (3分) (2)处应填：_____

36. (2分) (3)处应填：_____

37. (3分) (4)处应填：_____

38. (3分) (5)处应填：_____

(2)

（交朋友）根据社会学研究表明，人们都喜欢找和自己身高相近的人做朋友。现在有n名身高两两不相同的同学依次走入教室，调查人员想预测每个人在走入教室的瞬间最想和已经进入教室的哪个人做朋友。当有两名同学和这名同学的身高差一样时，这名同学会更想和高的那个人做朋友。比如一名身高为1.80米的同学进入教室时，有一名身高为1.79米的同学和一名

身高为1.81米的同学在教室里，那么这名身高为1.80米的同学会更想和身高为1.81米的同学做朋友。对于第一个走入教室的同学我们不做预测。由于我们知道所有人的身高和走进教室的次序，所以我们可以采用离线的做法来解决这样的问题，我们用排序加链表的方式帮助每一个人找到在他之前进入教室的并且和他身高最相近的人。（第一空2分，其余3分）

```
01 #include <iostream>
02 using namespace std;
03 #define MAXN 200000
04 #define infinity 2147483647
05 int answer[MAXN], height[MAXN], previous[MAXN], next_pos[MAXN];
06 int order[MAXN];
07 int n;
08 void sort(int l, int r)
09 {
10     int x = height[order[(l + r) / 2]], i = l, j = r, temp;
11     while (i <= j)
12     {
13         while (height[order[i]] < x)
14             i++;
15         while (height[order[j]] > x)
16             j--;
17         if (__(1)__)
18         {
19             temp = order[i];
20             order[i] = order[j];
21             order[j] = temp;
22             i++;
23             j--;
24         }
25     }
26     if (i < r)
27         sort(i, r);
28     if (l < j)
29         sort(l, j);
30 }
31 int main()
32 {
33     cin >> n;
34     int i, higher, shorter;
35     for (i = 1; i <= n; i++)
36     {
37         cin >> height[i];
```

```
38          order[i] = i;
39      }
40      sort(1, n);
41      for (i = 1; i <= n; i++)
42      {
43          previous[order[i]] = order[i - 1];
44          __(2)__;
45      }
46      for (i = n; i >= 2; i--)
47      {
48          higher = shorter = infinity;
49          if (previous[i] != 0)
50              shorter = height[i] - height[previous[i]];
51          if (next_pos[i] != 0)
52              __(3)__;
53          if (__(4)__)
54              answer[i] = previous[i];
55          else
56              answer[i] = next_pos[i];
57          next_pos[previous[i]] = next_pos[i];
58          __(5)__;
59      }
60      for (i = 2; i <= n; i++)
61          cout << i << ":" << answer[i];
62      return 0;
63  }
```

39. (2.8分) (1)处应填：_____

40. (2.8分) (2)处应填：_____

41. (2.8分) (3)处应填：_____

42. (2.8分) (4)处应填：_____

43. (2.8分) (5)处应填：_____

CSP-J 初赛模拟卷（第 14 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

1．一个 32 位整型变量占用（ ）个字节。

A．4　　B．8　　C．32　　D．128

2．二进制数 00100100 和 00010101 的和是（ ）。

A．00101000　　B．001010100　　C．01000101　　D．00111001

3．关于汇编语言，下列说法错误的是（ ）。

A．是一种与具体硬件相关的程序设计语言
B．在编写复杂程序时，相对于高级语言而言代码量较大，且不易调试
C．可以直接访问寄存器、内存单元、以及I/O端口
D．随着高级语言的诞生，如今已完全被淘汰，不再使用

4．FTP 可以用于（ ）。

A．远程传输文件　　B．发送电子邮件　　C．浏览网页　　D．网上聊天

5．原字符串中任意一段连续的字符所组成的新字符串称为子串。 则字符“AAABBBCCC”共有（ ）个不同的非空子串。

A．3　　B．12　　C．36　　D．45

6．设变量x为非负float型且已赋值，则以下语句中能将x中的数值保留到小数点后两位，并将第三位四舍五入的是（ ）。

A．x = (x * 100) + 0.5 / 100.0;　　B．x = (x * 100 + 0.5) / 100.0;
C．x = (int)(x * 100 + 0.5)/100.0;　　D．x = (x / 100 + 0.5) * 100.0;

7．一个字长为8位的整数的补码是11111001，则它的原码是（ ）。

A．00000111　　B．01111001　　C．11111001　　D．10000111

8．下面是根据欧几里得算法编写的函数，它所计算的是a和b的（ ）。 int euclid(int a, int b) { if (b == 0) return a; else return euclid(b, a % b); }

A．最大公共质因子　　B．最小公共质因子　　C．最大公约数　　D．最小公倍数

9．有7个一模一样的苹果，放到3个一样的盘子中，盘子可以为空，一共有（ ）种放法。

A．7　　B．8　　C．21　　D．3^7

10. 如果树根算第1层，那么一棵 n 层的二叉树最多有（ ）个结点。

A. 2^n-1　　B. 2^n　　C. 2^n+1　　D. 2^{n+1}

11. 无向完全图是图中每对顶点之间都恰好有一条边的简单图。已知无向完全图G有7个顶点，则它共有（ ）条边。

A. 7　　B. 21　　C. 42　　D. 49

12. 如果根的高度为 1,具有 61 个结点的完全二叉树的高度为（ ）。

A. 5　　B. 6　　C. 7　　D. 8

13. 对于给定的序列 $\{a_k\}$，我们把 (i,j) 称为逆序对当且仅当 $i<j$ 且 $a_i>a_j$。那么序列 $\{1,7,2,3,5,4\}$ 的逆序对数为（ ）个。

A. 4　　B. 5　　C. 6　　D. 7

14. 给定含有 n 个互不相同元素的单峰数组 L，满足先严格递增、后严格递减，峰顶位于数组内部。用下面的二分算法寻找峰顶，请将 a、b、c 三行代码依次填入空白处。正确的填空顺序是（ ）。

```
a. return Search(k + 1, r)
b. return Search(l, k - 1)
c. return L[k]

Search(l, r):
    if l == r then return L[l]
    k = floor((l + r) / 2)
    if L[k] > L[k - 1] and L[k] > L[k + 1]: ______
    else if L[k] < L[k + 1]: ______
    else: ______

调用 Search(2, n - 1)
```

A. c,a,b　　B. c,b,a　　C. a,b,c　　D. b,a,c

15. 广度优先搜索时，需要用到的数据结构是（ ）。

A. 链表　　B. 队列　　C. 栈　　D. 散列表

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 **√**，错误填 **×**；共 **18** 小题，共计 **37** 分，各小题分值见题目）

(1)

```
01 #include<iostream>
02 using namespace std;
03 #define MAXN 20
04 int gu[MAXN][MAXN];
05 int luo(int n, int m) {
06     if(n <= 1 || m < 2)
07         return 1;
08     if(gu[n][m] != -1)
09         return gu[n][m];
10     int ans=0;
11     for(int i = 0; i <m; i += 2)
12         ans+=luo(n-1, i);
13     gu[n][m] = ans;
14     return ans;
15 }
16 int main() {
17     int n, m;
18     cin >> n >> m;
19     for(int i = 0; i < MAXN; i++)
20         for(int j = 0; j < MAXN; j++)
21             gu[i][j]= -1;
22     cout << luo(n, m);
23     return 0;
24 }
```

判断题

16. (1分) luo函数中，m的值不可能是奇数。（ ）

17. (1分) 若将第11行的”<“改为”<=“,程序的输出结果可能会改变。（ ）

18. (2分) 若将第8、9、13行删除，程序的运行的结果不变。（ ）

19. (2分) 在添加合适的头文件后，将第19到21行替换为“memset(gu, 255,sizeof(gu))；”可以起到相同的作用。（ ）

单选题

20. (4分) 若输入数据为4 8，则输出为（ ）。

A. 7　　B. 8　　C. 15　　D. 16

21. (3分) 最坏情况下，此程序的时间复杂度是（ ）。

A. O(m²n)　　B. O(nm!)　　C. O(n²)　　D. O(n²m)

(2)

```
01 #include <iostream>
02 using namespace std;
03
04 long long n, ans;
05 int k, len;
06 long long d[1000000];
07
08 int main() {
09     cin >> n >> k;
10     d[0] = 0;
11     len= 1;
12     ans = 0;
13     for (long long i = 0; i <n; ++i) {
14         ++d[0];
15         for (int j = 0; j + 1<len; ++j) {
16             if (d[j] == k) {
17                 d[j] = 0;
18                 d[j + 1] += 1;
19                 ++ans;
20             }
21         }
22         if (d[len - 1] == k) {
23             d[len - 1] = 0;
24             d[len] =1;
25             ++len;
26             ++ans;
27         }
28     }
29     cout << ans << endl;
30     return 0;
31 }
```

假设输入的 n 是不超过 2^{62} 的正整数，k 都是不超过10000的正整数，完成下面的判断题和单选题：

判断题

22. (1.5分) 若 k = 1，则输出 ans 时，len = n。（ ）

23. (1.5分) 若 k > 1，则输出 ans 时，len 一定小于 n。（ ）

24. (1.5分) 若 k > 1，则输出 ans 时， k^{len} 一定大于 n。（ ）

单选题

25. (1.5分) 若输入的 n 等于:10^{15}，输入的 k 为1，则输出等于（ ）。

A. 1　　B. $(10^{30}-10^{15})/2$　　C. $(10^{30}+10^{15})/2$　　D. 10^{15}

26. (3分) 若输入的 n 等于205,891,132,094,649（即 3^{30}），输入的 k 为3，则输出等于（ ）。

A. 3^{30}　　B. $(3^{30}-1)/2$　　C. $3^{30}-1$　　D. $(3^{30}+1)/2$

27. (3分) 若输入的 n 等于100,010,002,000,090，输入的 k 为10，则输出等于（ ）。

A. 11,112,222,444,543　　B. 11,122,222,444,453

C. 11,122,222,444,543　　D. 11,112,222,444,453

(3)

```
01 #include<cstdio>
02 using namespace std;
03 int n, m;
04 int a[100], b[100];
05
06 int main() {
07     scanf("%d%d", &n, &m);
08     for (int i = 1; i <= n; ++i)
09         a[i] = b[i] = 0;
10     for (int i = 1; i <= m; ++i) {
11         int x, y;
12         scanf("%d%d", &x, &y);
13         if (a[x] < y && b[y] < x) {
14             if (a[x] > 0)
15                 b[a[x]] = 0;
16             if (b[y] > 0)
17                 a[b[y]] = 0;
18             a[x] = y;
19             b[y] = x;
20         }
21     }
22     int ans = 0;
23     for (int i = 1; i <= n; ++i) {
```

```
24          if (a[i] == 0)
25              ++ans;
26          if (b[i] == 0)
27              ++ans;
28      }
29      printf("%d", ans);
30      return 0;
31  }
```

假设输入的 n 和 m 都是正整数，x 和 y 都是在 $[1,n]$ 的范围内的整数，完成下面的判断题和单选题：

判断题

28. (1.5分) 当 m > 0 时，输出的值一定小于 2n。（ ）

29. (1.5分) 执行完第27行的”++ans”时，ans一定是偶数。（ ）

30. (1.5分) $a[i]$ 和 $b[i]$ 不可能同时大于0。（ ）

31. (1.5分) 右程序执行到第13行时，x总是小于y，那么第15行不会被执行。（ ）

单选题

32. (3分) 若 m 个 x 两两不同，且 m 个 y 两两不同，则输出的值为（ ）

A. 2n - 2m　　B. 2n + 2　　C. 2n - 2　　D. 2n

33. (3分) 若 m 个 x 两两不同，且 m 个 y 都相等，则输出的值为（ ）

A. 2n - 2　　B. 2n　　C. 2m　　D. 2n - 2m

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **10** 小题，共计 **29** 分，各小题分值见题目）

(1)

完善程序：

（打印日历）输入月份 $m(1 \leq m \leq 12)$，按一定格式打印2015年第 m 月的月历。例如，2015年1月的月历打印效果如下(第一列为周日)：

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

```
01 #include <iostream>
02 #include <string>
03 using namespace std;
04 const int dayNum[] = {-1, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31,
   30, 31};
05 int m, offset, i;
06 int main()
07 {
08     cin >> m;
09     cout << "S\tM\tT\tW\tT\tF\tS" << endl; /* '\t'为 TAB 制表符 */
10     __(1)__;
11     for (i = 1; i < m; i++)
12         offset = __(2)__;
13     for (i = 0; i < offset; i++)
14         cout << '\t';
15     for (i = 1; i <= __(3)__; i++)
16     {
17         cout << __(4)__;
18         if (i == dayNum[m] || __(5)__ == 0)
19             cout << endl;
20         else
21             cout << '\t';
22     }
23     return (0);
24 }
```

34. (3分) (1)处应填：_____

35. (3分) (2)处应填：_____

36. (2.5分) (3)处应填：_____

37. (2.5分) (4)处应填：_____

38. (3分) (5)处应填：_____

(2)

（矩阵变幻）有一个奇幻的矩阵，在不停的变幻，其变幻方式为： 数字 0 变成矩阵

```
0 0
0 1
```

数字 1 变成矩阵

```
1 1
1 0
```

最初该矩阵只有一个元素 0，变幻n 次后，矩阵会变成什么样？ 例如，矩阵最初为：[0]；矩阵变幻 1 次后：

```
0 0
0 1
```

矩阵变幻 2 次后：

```
0 0 0 0
0 1 0 1
0 0 1 1
0 1 1 0
```

输入一行一个不超过 10 的正整数 n。输出变幻 n 次后的矩阵。

试补全程序。

提示：

<< 表示二进制左移运算符，例如$(11)_2 << 2 = (1100)_2$；

而 ^ 表示二进制异或运算符，它将两个参与运算的数中的每个对应的二进制位进行比较，若两个二进制位相同，则运算结果的对应二进制位为 0，反之为 1。

```
01 #include <cstdio>
02 using namespace std;
03 int n;
04 const int max_size = 1 << 10;
05
06 int res[max_size][max_size];
07
08 void recursive(int x, int y, int n, int t) {
09     if (n == 0) {
10         res[x][y] = __(1)__;
```

```
        return;
    }
    int step = 1 << (n - 1);
    recursive(__(2)__, n - 1, t);
    recursive(x, y + step, n - 1, t);
    recursive(x + step, y, n - 1, t);
    recursive(__(3)__, n - 1, !t);
}

int main() {
    scanf("%d", &n);
    recursive(0, 0, __(4)__);
    int size = __(5)__;
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++)
            printf("%d", res[i][j]);
        puts("");
    }
    return 0;
}
```

39. (3分) (1)处应填

A. n%2　　B. 0　　C. t　　D. 1

40. (3分) (2)处应填

A. x-step,y-step　B. x,y-step　C. x-step,y　D. x, y

41. (3分) (3)处应填

A. x-step,y-step　B. $x+$ step, $y+$ step　C. x-step,y　D. x,y-step

42. (3分) (4)处应填

A. n-1,n%2　　B. n, 0　　C. n, n%2　　D. n-1, 0

43. (3分) (5)处应填

A. $1 << (n+1)$　　B. 1<<n　　C. $n+1$　　D. $1 \ll (n-1)$

CSP-J 初赛模拟卷（第 15 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

1．以下奖项与计算机领域最相关的是（　）。

A．奥斯卡奖　　B．图灵奖　　C．诺贝尔奖　　D．普利策奖

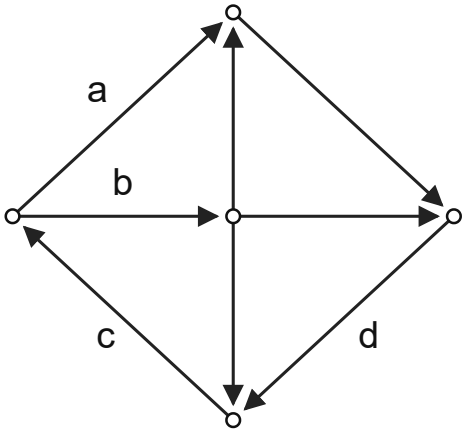
2．计算机应用的最早领域（　）

A．数值计算　　B．人工智能　　C．机器人　　D．过程控制

3．如果树根算第1层，那么一棵 n 层的二叉树最多有（　）个结点。

A．2^n-1　　B．2^n　　C．2^n+1　　D．2^{n+1}

4．对一个有向图而言，如果每个节点都存在到达其他任何节点的路径，那么就称它是强连通图。例如，下图就是一个强连通图。事实上，在删除边（　）后，它依然是强连通图。



A．a　　B．b　　C．c　　D．d

5．有以下程序：

```
#include <iostream>
using namespace std;
int main()
{
    int s, a, n;
    s= 0;
    a= 1;
    cin >> n;
    do
    {
        s+= 1;
        a-= 2;
    }
    while ( a != n );
    cout << s << endl;
    return(0);
}
```

若要使程序的输出值为2,则应该从键盘给*n*输入的值是（ ）。

A．-1　　B．-3　　C．-5　　D．0

6．对于给定的序列 $\{a_k\}$，我们把 (i,j) 称为逆序对当且仅当 $i<j$ 且 $a_i>a_j$。那么序列 $\{1,7,2,3,5,4\}$ 的逆序对数为（ ）个。

A．4　　B．5　　C．6　　D．7

7．以下关于字符串的判定语句中正确的是（ ）。

A．字符串是一种特殊的线性表　　B．串的长度必须大于零

C．字符串不可以用数组来表示　　D．空格字符组成的串就是空串

8．10000 以内，与 10000 互质的正整数有（ ）个。

A．2000　　B．4000　　C．6000　　D．8000

9．前序遍历序列与中序遍历序列相同的二叉树为（ ）。

A．根结点无左子树

B．根结点无右子树

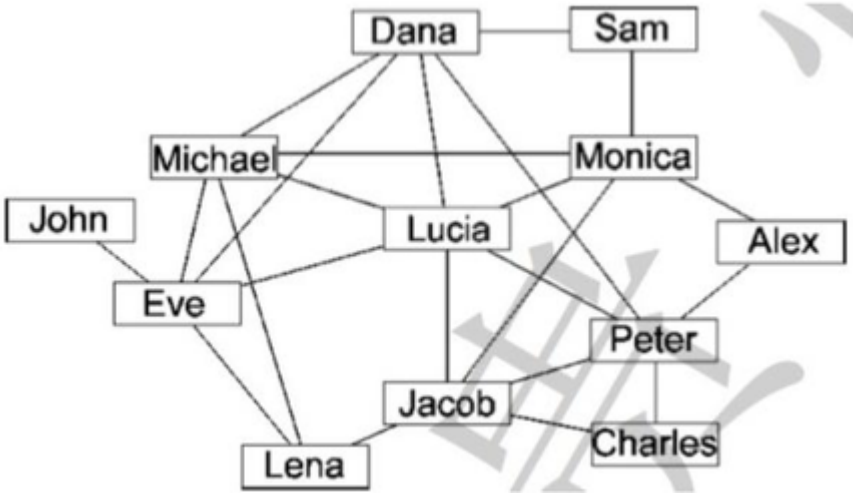
C．只有根结点的二叉树或非叶子结点只有左子树的二叉树

D．只有根结点的二叉树或非叶子结点只有右子树的二叉树

10. 设含有10个元素的集合的全部子集数为 S ，其中由7个元素组成的子集数为 T ，则 T/S 的值为（ ）。

A. 5 / 32　　B. 15 / 128　　C. 1 / 8　　D. 21 / 128

11. Lucia 和她的朋友以及朋友的朋友都在某社交网站上注册了账号。下图是他们之间的关系图，两个人之间有边相连接这两个人是朋友，没有边相连代表不是朋友。这个社交网站的规则是:如果某人人向他（她）的朋友 B 分享了某张照片，那么 B 就可以对该照片进行评论；如果 B 评论了该照片，那么他（她）的所有朋友都可以看见这个评论以及被评论的照片，但是不能对该照片进行评论（除非 A 也向他（她）分享了该照片）。现在 Lucia 已经上传了一张照片，但是她不想让 Jacob 看见这张照片，那么她可以向以下朋友（ ）分享该照片。



A. Dana, Michael, Eve　　B. Dana, Eve, Monica

C. Michael, Eve, Jacob　　D. Micheal, Peter, Monica

12. 为了统计一个非负整数的二进制形式中 1 的个数，代码如下： int CountBit(int x) { int ret = 0; while (x) { ret++; _____; } return ret; } 则空格内要填入的语句是（ ）。

A. x >>= 1　　B. x &= x - 1　　C. x |= x >> 1　　D. x <<= 1

13. 新学期开学了，小胖想减肥，健身教练给小胖制定了两个训练方案。方案一：每次连续跑3公里可以消耗300千卡（耗时半小时）；方案二：每次连续跑5公里可以消耗600千卡（耗时1小时）。小胖每周周一到周四能抽出半小时跑步，周五到周日能抽出一小时跑步。另外，教练建议小胖每周最多跑21公里，否则会损伤膝盖。 请问如果小胖想严格执行教练的训练方案，并且不想损伤膝盖，每周最多通过跑步消耗多少千卡？（ ）

A. 3000　　B. 2500　　C. 2400　　D. 2520

14. 有 10 个顶点的无向图至少应该有（ ）条边才有可能构成连通图。

A. 9　　B. 10　　C. 11　　D. 12

15. Linux下可执行文件的默认扩展名为（ ）。

A. exe　　B. com　　C. dll　　D. 以上都不是

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 **√**，错误填 **×**；共 **12** 小题，共计 **27** 分，各小题分值见题目）

(1)

```
01 #include<iostream>
02 using namespace std;
03 #define MAXN 20
04 int gu[MAXN][MAXN];
05 int luo(int n, int m) {
06     if(n <= 1 || m < 2)
07         return 1;
08     if(gu[n][m] != -1)
09         return gu[n][m];
10     int ans=0;
11     for(int i = 0; i <m; i += 2)
12         ans+=luo(n-1, i);
13     gu[n][m] = ans;
14     return ans;
15 }
16 int main() {
17     int n, m;
18     cin >> n >> m;
19     for(int i = 0; i < MAXN; i++)
20         for(int j = 0; j < MAXN; j++)
```

```
21              gu[i][j]= -1;
22      cout << luo(n, m);
23      return 0;
24  }
```

判断题

16. (1分) luo函数中，m的值不可能是奇数。（ ）

17. (1分) 若将第11行的“<”改为“<=”，程序的输出结果可能会改变。（ ）

18. (2分) 若将第8、9、13行删除，程序的运行的结果不变。（ ）

19. (2分) 在添加合适的头文件后，将第19到21行替换为“memset(gu, 255,sizeof(gu));”可以起到相同的作用。（ ）

单选题

20. (4分) 若输入数据为4 8，则输出为（ ）。

A. 7　　B. 8　　C. 15　　D. 16

21. (3分) 最坏情况下，此程序的时间复杂度是（ ）。

A. $O(m^2n)$　　B. O(nm!)　　C. $O(n^2)$　　D. $O(n^2m)$

(2)

```
01  #include <stdio.h>
02  #include <string.h>
03
04  char base[64];
05  char table[256];
06  char str[256];
07  char ans[256];
08
09  void init()
10  {
11      for (int i = 0; i < 26; i++) base[i] = 'A' + i;
12      for (int i = 0; i < 26; i++) base[26 + i] = 'a' + i;
13      for (int i = 0; i < 10; i++) base[52 + i] = '0' + i;
14      base[62] = '+', base[63] = '/';
15
16      for (int i = 0; i < 256; i++) table[i] = 0xff;
17      for (int i = 0; i < 64; i++) table[base[i]] = i;
18      table['='] = 0;
```

```
}

void decode(char *str)
{
    char *ret = ans;
    int i, len = strlen(str);
    for (i = 0; i < len; i += 4) {
        (*ret++) = table[str[i]] << 2 | table[str[i + 1]] >> 4;
        if (str[i + 2] != '=')
            (*ret++) = (table[str[i + 1]] & 0x0f) << 4 | table[str[i + 2]] >> 2;
        if(str[i + 3] != '=')
            (*ret++) = table[str[i + 2]] << 6 | table[str[i + 3]];
    }
}

int main()
{
    init();
    printf("%d\n", (int)table[0]);

    scanf("%s", str);
    decode(str);
    printf("%s\n", ans);
    return 0;
}
```

假设 char 是 8 位有符号类型，输入字符串长度为 4 的倍数，且输入不超过数组范围。

判断题

22. (1.5分) 输出的第二行一定是由小写字母、大写字母、数字和 +、/、= 构成的字符串。（ ）

23. (1.5分) 可能存在输入不同，但输出的第二行相同的情形。（ ）

24. (1.5分) 输出的第一行为 -1。（ ）

单选题

25. (3分) 设输入字符串长度为 n，decode 函数的时间复杂度为（ ）。

A. $O(\sqrt{n})$　　B. $O(n)$　　C. $O(n\log n)$　　D. $O(n^2)$

26. (3分) 当输入为 Y3Nx 时，输出的第二行为（ ）。

A. csp　　B. csq　　C. CSP　　D. Csp

27. (3.5分) 当输入为 Y2NmIDIwMjE= 时，输出的第二行为（ ）。

A. ccf2021　　B. ccf2022　　C. ccf 2021　　D. ccf 2022

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **14** 小题，共计 **37** 分，各小题分值见题目）

(1)

(马走日)回溯算法实际上一个类似枚举的搜索尝试过程，主要是在搜索尝试过程中寻找问题的解，当发现已不满足求解条件时，就“回溯”返回，尝试其他路径。回溯法是一种选优搜索法按选向前索以达到目标。但当搜索到某一步时，若发现原先选择并优或达不到目标,则退回一步重新选择，这种走不通就退回去再走的技术称为回溯法，而满足回溯条件的某个状态的点称为回溯点。 马在中国象棋中以日字形规则移动。请编写一段程序，给定rxc大小的棋盘以及马的初始位置(m,n)，要求不能重复盘上的同一个点,计算马有多少途径可以遍历盘上的所有点。

```
01 #include<iostream>
02 using namespace std;
03 int r,c;
04 int cnt,tot;
05 int wayr[8]={2,2,1,-1,-2,-2,1,-1};
06 int wayc[8]={1,-1,2,2,1,-1,-2,-2};
07 bool mark[1001][1001];
08 bool check(int x, int y)
09 {
10     if(__(1)__) return true;
11     return false;
12 }
13 void search(int x,int y)
14 {
15     for(int i=0;i<8;i++)
16         if(__(2)__ && __(3)__)
17         {
18             mark[x+wayr[i]][y+wayc[i]]=true;
19             tot++;
20             if(__(4)__) cnt++;
21             search(__(5)__);
22             tot--;
```

```
23              mark[x+wayr[i]][y+wayc[i]]=false;
24          }
25  }
26  int main()
27  {
28      cnt=0;
29      int m,n;
30      cin>>r>>c>>m>>n;
31      if(!check(m,n)) cout<<0<<endl;
32      else if(r==1&&c==1) cout<<1<<endl;
33      else
34      {
35          mark[m][n]=true;
36          search(m,n);
37          cout<<cnt<<endl;
38      }
39  }
```

28. (3分) (1)处应填

A. x>=0 || y>=0 || x<r || y<c. B. x>=0 || y>=0 && x<r || y<e.
C. x>=0 && y>=0 && x<r && y<c D. x>=0&&y>=0 || x<r &&y<c

29. (3分) (2)处应填

A. check(wayr[i], wayc[i]) B. check(x+wayr[i], y+wayc[i])
C. !check(x+wayr[i],y+wayc[i]) D. !check(wayr[i],wayc[i])

30. (3分) (3)处应填

A. mark[wayr[i]][wayc[i]] B. mark[x+wayr[i]][y+wayc[i]]
C. lmark[wayr[i]][wayc[i]] D. !mark[x+wayr[i]][y+wayc[i]]

31. (3分) (4)处应填

A. tot==r * c-1 B. tot-=r * c C. cnt==r * c-1 D. cnt==r * c

32. (3分) (5)处应填

A. wayr[i], wayc[i] B. x-wayr[i], y-wayc[i]
C. x-wayr[i], y+wayc[i] D. $x+\mathrm{wayr}[i], y+\mathrm{wayc}[i]$

(2)

完善程序

（子矩阵）给输入一个 $n1 \times m1$ 的矩阵 a，和 $n2 \times m2$ 的矩阵 b，问 a 中是否存在子矩阵和 b 相等。若存在，输出所有子矩阵左上角的坐标:若不存在输出“There is no answer”。

```
01 #include<iostream>
02 using namespace std;
03 const int SIZE = 50;
04 int n1,m1,n2,m2,a[SIZE][SIZE],b[SIZE][SIZE];
05 int main()
06 {
07     int i,j,k1,k2;
08     bool good ,haveAns;
09     cin>>n1>>m1;
10     for(i=1;i<=n1;i++)
11         for(j=1;j<=m1;j++)
12             cin>>a[i][j];
13     cin>>n2>>m2;
14     for(i=1;i<=n2;i++)
15         for(j=1;j<=m2;j++)
16             __(1)__;
17     haveAns=false;
18     for(i=1;i<=n1-n2+1;i++)
19         for(j=1;j<= __(2)__;j++){
20             __(3)__;
21             for(k1=1;k1<=n2;k1++)
22                 for(k2=1;k2<=__(4)__ ;k2++){
23                     if(a[i+k1-1][j+k2-1]!=b[k1][k2])
24                         good=false;
25                 }
26             if(good){
27                 cout<<i<<' '<<j<<endl;
28                 __(5)__;
29             }
30         }
31     if(!haveAns)
32         cout<<"There is no answer"<<endl;
33     return 0;
34 }
```

33. (2分) (1)处应填：_____

34. (2分) (2)处应填：_____

35. (2分) (3)处应填：_____

36. (2分) (4)处应填：_____

37. (2分) (5)处应填：_____

(3)

完善程序： (数字删除)下面程序的功能是将字符串中的数字字符删除后输出。请填空。

```
#include <iostream>
using namespace std;
int delnum( char *s )
{
    int i, j;
    j = 0;
    for ( i = 0; s[i] != '\0'; i++ )
        if ( s[i] < '0'   __(1)__  s[i] > '9' )
        {
            s[j] = s[i];
            __(2)__;
        }
    return(__(3)__);
}
const int SIZE = 30;
int main()
{
    char s[SIZE];
    int len, i;
    cin.getline( s, sizeof(s) );
    len = delnum( s );
    for ( i = 0; i < len; i++ )
        cout << __(4)__;
    cout << endl;
    return 0;
}
```

38. (3分) (1)处应填：_____

39. (3分) (2)处应填：_____

40. (3分) (3)处应填：_____

41. (3分) (4)处应填：_____

CSP-J 初赛模拟卷（第 16 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

1．在 PC 机中，PENTIUM(奔腾)、酷睿、赛扬等是指（ ）。

A．生产厂家名称　　B．硬盘的型号　　C．CPU 的型号　　D．显示器的型号

2．1TB代表的字节数是（ ）。

A．2的10次方　　B．2的20次方　　C．2的30次方　　D．2的40次方

3．如果一个栈初始时为空，且当前栈中的元素从栈底到栈顶依次为 a,b,c ，另有元素 d 已经出栈，则可能的入栈顺序是（ ）。

A．a,d,c,b　　B．b,a,c,d　　C．a,c,b,d　　D．d,a,b,c

4．设 X 、Y、Z分别代表三进制下的一位数字，若等式 $XY+ZX=XYX$ 在三进制下成立，那么同样在三进制下，等式 $XY*ZX=()$ 也成立。

A．Y X Z　　B．Z X Y　　C．X Y Z　　D．X Z Y

5．一棵具有5层的满二叉树中结点数为（ ）。

A．31　　B．32　　C．33　　D．16

6．给定含有 n 个互不相同元素的单峰数组 L，满足先严格递增、后严格递减，峰顶位于数组内部。用下面的二分算法寻找峰顶，请将 a、b、c 三行代码依次填入空白处。正确的填空顺序是（ ）。

```
a. return Search(k + 1, r)
b. return Search(l, k - 1)
c. return L[k]

Search(l, r):
    if l == r then return L[l]
    k = floor((l + r) / 2)
    if L[k] > L[k - 1] and L[k] > L[k + 1]: ______
    else if L[k] < L[k + 1]: ______
    else: ______

调用 Search(2, n - 1)
```

A．c,a,b　　B．c,b,a　　C．a,b,c　　D．b,a,c

7．表达式 $a*(b+c)*d$ 的后缀形式是（ ）。

A．$abcd*+*$　　B．$abc+*d*$　　C．$a*bc+*d$　　D．$b+c*a*d$

8．现有一段文言文，要通过二进制哈夫曼编码进行压缩。简单起见，假设这段文言文只由4个汉字“之”、“乎”、“者”、“也”组成，它们出现的次数分别为700、600、300、400。若待合并结点权值相同，优先选取由其他结点合并形成的结点。那么，“也”字的编码长度可能是（ ）。

A．1　　B．2　　C．3　　D．4

9．假设一棵二叉树的后序遍历序列为 DGJHEBIFCA，中序遍历序列为DBGEHJACIF,则其前序遍历序列为（ ）。

A．ABCDEFGHIJ　　B．ABDEGHJCFI　　C．ABDEGJHCFI　　D．ABDEGHJFIC

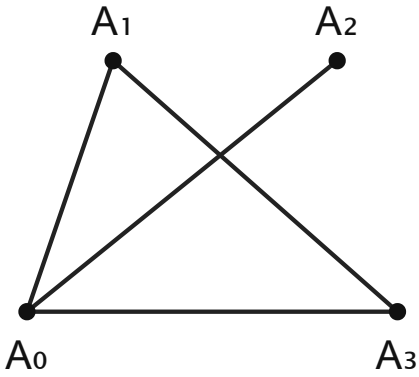
10．若有如下程序段，其中 s 、 a 、 b 、 c 均已定义为整型变量，且 a 、 c 均已赋值（c 大于0）。

$$\begin{aligned}&\mathrm{s=a;}\\&\mathrm{for\ (b=1;b<=c;\ b{+}{+})}\\&\mathrm{s=s+1;}\end{aligned}$$

则与上述程序段修改 s 值的功能等价的赋值语句是（ ）。

A．$s=a+b$　　B．$s=a+c$　　C．$s=s+c$　　D．$s=b+c$

11．以 A_0 作为起点，对下面的无向图进行深度优先遍历时，遍历顺序不可能是（　）。



A．A_0, A_1, A_2, A_3　　B．A_0, A_1, A_3, A_2　　C．A_0, A_2, A_1, A_3　　D．A_0, A_3, A_1, A_2

12．完全二叉树的顺序存储方案，是指将完全二叉树的结点从上至下、从左至右依次存放到一个顺序结构的数组中。假定根结点存放在数组的1号位置，则第k号结点的父结点如果存在的话，应当存放在数组的（　）号位置。

A．2k　B．$2k+1$　C．$k/2$ 下取整　D．$(k+1)/2$ 下取整

13．周末小明和爸爸妈妈三个人一起想动手做三道菜。小明负责洗菜、爸爸负责切菜、妈妈负责炒菜。假设做每道菜的顺序都是:先洗菜10分钟，然后切菜10分钟，最后炒菜10分钟。那么做一道菜需要30分钟。注意:两道不同的菜的相同步骤不可以同时进行。例如第一道菜和第二道的菜不能同时洗，也不能同时切。那么做完三道菜的最短时间需要（　）分钟。

A．90　　B．60　　C．50　　D．40

14．有向图中每个顶点的度等于该顶点的（　）。

A．入度　B．出度　C．入度和出度之和　D．入度和出度之差

15. 干支纪年法是中国传统的纪年方法，由10个天干和12个地支组合成60个天干地支。由公历年份可以根据以下公式和表格换算出对应的天干地支。 天干 = (公历年份) 除以10所得余数 地支 = (公历年份) 除以12所得余数 | 天干 | 甲 | 乙 | 丙 | 丁 | 戊 | 己 | 庚 | 辛 | 壬 | 癸 | | | | – | – | – | – | – | – | – | – | – | – | – | | | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | | | | 地支 | 子 | 丑 | 寅 | 卯 | 辰 | 巳 | 午 | 未 | 申 | 酉 | 戌 | 亥 | | | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 0 | 1 | 2 | 3 | 例如，今年是2020年，2020除以10余数为0，查表为“庚”；2020除以12余数为4，查表为“子”所以今年是庚子年。请问1949年的天干地支息{。

A. 己酉　　B. 己亥　　C. 己丑　　D. 己卯

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 √，错误填 ×；共 18 小题，共计 42 分，各小题分值见题目）

(1)

```
01 #include <iostream>
02 using namespace std;
03
04 int n;
05 int d[1000];
06
07 int main() {
08     cin >> n;
09     for (int i = 0; i < n; ++i)
10         cin >> d[i];
11     int ans = -1;
12     for (int i = 0; i < n; ++i)
13         for (int j = 0; j < n; ++j)
14             if (d[i] < d[j])
15                 ans = max(ans, d[i] + d[j] - (d[i] & d[j]));
16     cout << ans;
17     return 0;
18 }
```

假设输入的n和d[i]都是不超过10000的正整数，完成下面的判断题和单选题:

判断题

16. (1.5分) n必须小于1000，否则程序可能会发生运行错误。（　）

17. (1.5分) 输出一定大于等于0。（ ）

18. (1.5分) 若将第13行的“j = 0”改为“j = i + 1”，程序输出可能会改变。（ ）

19. (1.5分) 将第14行的“d[i] < d[j]”改为“d[i] != d[j]”，程序输出不会改变。（ ）

单选题

20. (3分) 若输入 n 为100，且输出为127，则输入的d[i]中不可能有（ ）。

A. 127　　B. 126　　C. 128　　D. 125

21. (3分) 若输出的数大于0，则下面说法正确的是（ ）。

A. 若输出为偶数，则输入的d[i]中最多有两个偶数
B. 若输出为奇数，则输入的d[i]中至少有两个奇数
C. 若输出为偶数，则输入的d[i]中至少有两个偶数
D. 若输出为奇数，则输入的d[i]中最多有两个奇数

(2)

```
01 #include <iostream>
02 using namespace std;
03 const int maxn = 10000;
04 int n;
05 int a[maxn];
06 int b[maxn];
07 int f(int l, int r, int depth) {
08     if (l > r)
09         return 0;
10     int min = a[l], mink = l;
11     for (int i = l; i <= r; ++i) {
12         if (min > a[i]) {
13             min = a[i];
14             mink = i;
15         }
16     }
17     int lres = f(l, mink - 1, depth + 1);
18     int rres = f(mink + 1, r, depth + 1);
19     return lres + rres + depth * b[mink];
20 }
21 int main() {
22     cin >> n;
```

```
    for (int i = 0; i < n; ++i)
        cin >> a[i];
    for (int i = 0; i < n; ++i)
        cin >> b[i];
    cout << f(0, n - 1, 1) << endl;
    return 0;
}
```

判断题

22. (1.5分) 如果a数组有重复的数字，则程序运行时会发生错误。（ ）

23. (1.5分) 如果b数组全为0，则输出为0。（ ）

单选题

24. (3分) 当 $n = 100$ 时，最坏情况下，与第12行的比较运算执行的次数最接近的是:（ ）。

A. 5000　　B. 600　　C. 6　　D. 100

25. (3分) 当n=100时，最好情况下，与第12行的比较运算执行的次数最接近的是:（ ）。

A. 100　　B. 6　　C. 5000　　D. 600

26. (3分) 当 $n = 10$ 时，若 b 数组满足，对任意 $0 <= i < n$, 都有 $b[i] = i + 1$ ，那么输出最大为（ ）。

A. 386　　B. 383　　C. 384　　D. 385

27. (4分) 当 $n = 100$ 时，若 b 数组满足，对任意 $0 <= i < n$, 都有 $b[i] = 1$ ，那么输出最小为（ ）。

A. 582　　B. 580　　C. 579　　D. 581

(3)

```
#include<bits/stdc++.h>
using namespace std;
#define MOD 19260817
#define MAXN 1005
long long A[MAXN][MAXN]={0},sum[MAXN][MAXN]={0};
int n, m, q;
int main(){
    A[1][1]=A[1][0]=1;
```

```
09      for(int i=2;i <= 1000; i++) {
10          A[i][0]=1;
11          for(int j = 1; j <= i; j++)
12              A[i][j]=(A[i-1][j]+A[i-1][j-1])%MOD;
13      }
14      for(int i =1;i <= 1000; i++)
15          for(int j=1;j<=1000;j++)
16              sum[i][j]=(sum[i-1][j]+sum[i][j-1]
17              -sum[i-1][j-1]+A[i][j]+MOD)%MOD;
18              int q;
19      cin >> q;
20      while(q--){
21          int n, m;
22          cin >> n >> m;
23          cout << sum[n][m] << endl;
24      }
25      return 0;
26  }
```

判断题

28. (1分) 当i<=j时，A[i][i]的值是0。（ ）

29. (2分) 当i>j时，A[i][j]的值相当于从i个不同元素中取出j个元素的排列数。（ ）

30. (2分) sum[i][j]的值(1<j<i≤1000)不小于sum[i-1][j-1]的值。（ ）

31. (2分) 若将第12行改为“A[i][j]=(A[i-1][j]+A[i-1][j-1]+MOD)%MOD;”，程序的运行的结果不变。（ ）

单选题

32. (4分) A[i][j] (1≤i≤10, 1≤j≤10)的所有元素中，最大值是（ ）。

A. 126　　B. 276　　C. 252　　D. 210

33. (3分) 若输入数据为1/5 3(其中“/”为换行符)，则输出为（ ）。

A. 10　　B. 35　　C. 50　　D. 24

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **10** 小题，共计 **28** 分，各小题分值见题目）

(1)

完善程序:

（读入整数）请完善下面的程序，使得程序能够读入两个int范围内的整数，并将这两个整数分别输出，每行一个。输入的整数之间和前后只会出现空格或者回车。输入数据保证合法。

例如:

输入:

```
123 -789
```

输出:

```
123
-789
```

```
#include <iostream>
using namespace std;
int readint(){
    int num = 0;          // 存储读取到的整数
    int negative = 0;    // 负数标识
    char c;               // 存储当前读取到的字符
    c = cin.get();
    while ((c < '0' || c > '9') && c != '-')
        c = __(1)__;
    if (c == '-')
        negative = 1;
    else
        __(2)__;
    c = cin.get();
    while (__(3)__){
        __(4)__;
        c = cin.get();
    }
    if (negative == 1)
        __(5)__;
    return num;
}
```

```
int main()
{
    int a, b;
    a = readint();
    b = readint();
    cout << a << endl
         << b << endl;
    return 0;
}
```

34. (2.5分) (1)处应填：_____

35. (3分) (2)处应填：_____

36. (3分) (3)处应填：_____

37. (3分) (4)处应填：_____

38. (2.5分) (5)处应填：_____

(2)

完善程序：

(中位数 median) 给定 n(n 为奇数且小于 1000)个整数，整数的范围在 $0-m(0<m<2^{31})$ 之间，请使用二分法求这 n 个整数的中位数。所谓中位数，是指将这 n 个排序之后，排在正中间的数。

```
#include <iostream>
using namespace std;
const int MAXN = 1000;
int n, i, lbound, rbound, mid, m, count;
int x[MAXN];
int main()
{
    cin >> n >> m;
    for (i = 0; i < n; i++)
        cin >> x[i];
    lbound = 0;
    rbound = m;
    while (__(1)__)
    {
        mid = lbound + (rbound - lbound) / 2;
        __(2)__;
```

```
17          for (i = 0; i < n; i++)
18              if (__(3)__)
19                  __(4)__;
20          if (count > n / 2)
21              lbound = mid + 1;
22          else
23              __(5)__;
24          cout << mid << " " << lbound << " " << rbound << " " <<
    count << endl;
25      }
26      cout << rbound << endl;
27      return (0);
28  }
```

39. (3分) (1)处应填：_____

40. (3分) (2)处应填：_____

41. (3分) (3)处应填：_____

42. (3分) (4)处应填：_____

43. (2分) (5)处应填：_____

CSP-J 初赛模拟卷（第 17 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

1．十进制数 114 的相反数的8位二进制补码是：

A．10001110　　B．10001101　　C．01110010　　D．01110011

2．以下哪个网站不是 Online Judge（在线程序判题系统）？Online Judge 可以查看算法题目，提交自己编写的程序，然后可以获得评测机反馈的结果。

A．Luogu　　B．Gitee　　C．LeetCode　　D．Codeforces

3．小A用字母A表示1，B表示2，以此来推，用 Z 表示 26。对于27以上的数，可以用两位或者更长的字符串来对应，例如AA对应27，AB 对应 28,AZ 对应52，AAA 对应703……那么BYT 字符串对应的数字是什么？

A．2018　　B．2020　　C．2022　　D．2024

4．UIM 拍摄了一张照片，其分辨率是 4096× 2160，每一个像素都是 24位真彩色。在没有压缩的情况下，这张图片占用空间接近以下哪个值？

A．8MB　　B．25MB　　C．200MB　　D．200KB

5．在一个长度为n的数组中找到第k大的数字，平均的算法时间复杂度最低的是

A．O (n)　　B．O (nk)　　C．O (nlogn)　　D．O (n^2)

6．对于“树”这种数据结构，正确的有： ①一个有n个顶点、n-1 条边的图是树 ②一个树中的两个顶点之间有且只有一条简单路径 ③树中一定存在度数不大于 1的顶点 ④树可能存在环

A．①②④　　B．①②③　　C．②③　　D．①②

7．博艾中学进行了一次信息学会考测试，其优、良、及格、不及格的试卷数量分别为10、13、14、5张。现在这些卷子混在一起，要将这些卷子按照等级分为4叠。分卷子的方法是，每次将一叠有不同等级答卷的卷子分为两堆，使得这两堆中没有相同等级的卷子，然后可以再分，直到分为4叠。要分完这些卷子，至少需要多少次“分卷子”的操作？将一堆数量为n的卷子分成两堆，就会产生n次分卷子的操作。

A．84　　B．93　　C．78　　D．85

8．一个二叉树的前序遍历是HGBDAFEC，中遍历是BGHFAEDC，同时采用顺序存储结构，即用一维数组元素存储该二叉树中的结点(根结点的下标为1，若某结点的下标为i，则其左孩子位于下标2i处、右孩子位于下标2i+1处)，则该数组的最大下标至少为（ ）

A．7　　B．13　　C．15　　D．12

9．在C++语言中，如果a=1;b=0;c=1;那么以下表达式中为1的是:

A．a&&b||b&&c　B．a+b>c||b　C．!(!c&&(!a||b))　D．a+b+c

10．在一个初始长度为n的链表中连续进行k次操作，每次操作是读入两个数字a_i和b_i，在链表中找到元素为a_i的结点(假设一定可以找到)，然后将这b_i个元素插入到这个结点前面。在最理想的情况下，链表访问的结点数量最少可能是多少(不算将要插入的结点)?

A．n次　　B．k次　　C．nk次　　D．n+k次

11．A班有5名风纪委员，B班有4名风纪委员，C班有3名风纪委员。现在需要这些同学中选取6名风纪委员巡逻，如果只关注各班派出的风纪委员人数，有几种不同的方案?

A．9　　B．12　　C．15　　D．18

12．以下哪种排序算法的时间复杂度是O(n^2)?

A．计数排序　　B．插入排序　　C．希尔排序　　D．归并排序

13．已知rand()可以生成一个0到32767的随机整数，如果希望得到一个范围在[a,b)的随机整数，a和b均是不超过100的正整数且a<b，那么可行的表达式是什么?

A．(rand()%(b-a))+a　　B．(rand()%(b-a+1))+a
C．(rand()%(b-a))+a+1　D．(rand()%(b-a+1))+a+1

14．一个7个顶点的完全图需要至少删掉多少条边才能变为森林?

A．16　　B．21　　C．15　　D．6

15．2020年8月，第（ ）届全国青少年信息学奥林匹克竞赛在（ ）举行?

A．26，广州　　B．26，长沙　　C．37，广州　　D．37，长沙

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 **√**，错误填 **×**；共 **18** 小题，共计 **40** 分，各小题分值见题目）

(1)

```
01 #include<iostream>
02 using namespace std;
03 #define MAXN 20
04 int gu[MAXN][MAXN];
05 int luo(int n, int m) {
06     if(n <= 1 || m < 2)
07         return 1;
08     if(gu[n][m] != -1)
09         return gu[n][m];
10     int ans=0;
11     for(int i = 0; i <m; i += 2)
12         ans+=luo(n-1, i);
13     gu[n][m] = ans;
14     return ans;
15 }
16 int main() {
17     int n, m;
18     cin >> n >> m;
19     for(int i = 0; i < MAXN; i++)
20         for(int j = 0; j < MAXN; j++)
21             gu[i][j]= -1;
22     cout << luo(n, m);
23     return 0;
24 }
```

判断题

16. (1分) luo函数中，m的值不可能是奇数。（ ）

17. (1分) 若将第11行的“<”改为“<=”，程序的输出结果可能会改变。（ ）

18. (2分) 若将第8、9、13行删除，程序的运行的结果不变。（ ）

19. (2分) 在添加合适的头文件后，将第19 到 21 行替换为“memset(gu, 255,sizeof(gu));”可以起到相同的作用。（ ）

单选题

20. (4分) 若输入数据为 4 8，则输出为（ ）。

A. 7　　B. 8　　C. 15　　D. 16

21. (3分) 最坏情况下，此程序的时间复杂度是（ ）。

A. $O(m^2n)$　　B. $O(mn!)$　　C. $O(n^2)$　　D. $O(n^2m)$

(2)

```
01 #include<cstdio>
02 #include<cstring>
03 using namespace std;
04 int n, m;
05 int f[101][101];
06 int F[101][101];
07 int main(){
08     scanf("%d%d",&n,&m);//n的值在1到100之间
09     memset(f,-1,sizeof(f));
10     for(int i = 1; i <=  m; i++) {
11         int u,v,w;//w的值在0到10000之间
12         scanf("%d%d%d",&u,&v,&w);
13         f[u][v]=f[v][u]=w;
14     }
15     for(int k = 1; k <= n; k++)
16         for(int i=1;i <=n; i++)
17             for(int j=1;j<= n; j++)
18                 if(f[i][k]!=-1&&f[k][j]!=-1)
19                     if(f[i][j]==-1||f[i][j]>f[k][j]+f[i][k])
20                         f[i][j]=f[i][k]+f[k][j];
21     int ans=2147483647;
22     for(int i=1;i<=n;i++)
23         for(int j=1;j<=n; j++){
24             for(int x=1;x<=n; x++)
25                 for(int y=1;y<=n; y++)
26                     F[x][y]=f[x][y];
27             F[i][j]=F[j][i]=0;
28             for(int x=1;x<= n; x++)
29                 for(int y=1;y<=n; y++)
30                     if(F[x][y]==-1||F[x][y]>F[x][i]+F[i][y])
31                         F[x][y]=F[x][i]+F[i][y];
32             for(int x=1;x<=n; x++)
33                 for(int y=1;y<=n; y++)
```

```
34                      if(F[x][y]==-1||F[x][y]>F[x][j]+F[j][y])
35                          F[x][y]=F[x][j]+F[j][y];
36              int res=0;
37              for(int x=1;x<=n; x++)
38                  for(int y=1;y<x; y++)
39                      res +=F[x][y];
40              ans = res < ans ? res : ans;
41          }
42      printf("%d\n",ans);
43      return 0;
44  }
```

判断题

22. (1分) 14到16行，将外层到内层的循环变量依次调整为i、j、k，程序的运行的结果不变。（ ）

23. (2分) 这个程序的时间复杂度和m无关。（ ）

24. (2分) 20行的ans如果初始化为10^7时，可能无法得到正确结果。（ ）

25. (2分) 若将第27到30行的部分和31到34行的两个部分互换，程序的运行的结果不变。（ ）

单选题

26. (3分) 若输入数据为4 5/1 2 3/1 3 6/2 3 4/2 4 7/3 4 2(其中“/”为换行符)，则输出为（ ）。

A. 14　　B. 18　　C. 21　　D. 28

27. (3分) 这个程序使用了（ ）算法。

A. Floyd　　B. Dijkstra　　C. Prim　　D. Kruskal

(3)

```
01  #include<bits/stdc++.h>
02  using namespace std;
03  #define MOD 19260817
04  #define MAXN 1005
05  long long A[MAXN][MAXN]={0},sum[MAXN][MAXN]={0};
06  int n, m, q;
07  int main(){
08      A[1][1]=A[1][0]=1;
```

```
09      for(int i=2;i <= 1000; i++) {
10          A[i][0]=1;
11          for(int j = 1; j <= i; j++)
12              A[i][j]=(A[i-1][j]+A[i-1][j-1])%MOD;
13      }
14      for(int i =1;i <= 1000; i++)
15          for(int j=1;j<=1000;j++)
16              sum[i][j]=(sum[i-1][j]+sum[i][j-1]
17              -sum[i-1][j-1]+A[i][j]+MOD)%MOD;
18              int q;
19      cin >> q;
20      while(q--){
21          int n, m;
22          cin >> n >> m;
23          cout << sum[n][m] << endl;
24      }
25      return 0;
26  }
```

判断题

28. (1分) 当i<=j时，A[i][i]的值是0。（ ）

29. (2分) 当i>j时，A[i][j]的值相当于从i个不同元素中取出j个元素的排列数。（ ）

30. (2分) sum[i][j]的值(1<j<i≤1000)不小于sum[i-1][j-1]的值。（ ）

31. (2分) 若将第12行改为“A[i][j]=(A[i-1][j]+A[i-1][j-1]+MOD)%MOD;”，程序的运行的结果不变。（ ）

单选题

32. (4分) A[i][j]（1≤i≤10，1≤j≤10)的所有元素中，最大值是（ ）。

A. 126　　B. 276　　C. 252　　D. 210

33. (3分) 若输入数据为1/5 3(其中“/”为换行符)，则输出为（ ）。

A. 10　　B. 35　　C. 50　　D. 24

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **10** 小题，共计 **30** 分，各小题分值见题目）

(1)

(封禁xxs)现有n个xxs(编号为1到n)，每个xxs都有一个关注者，第i个xxs的关注者是a_i。现在管理员要将其中的一些xxs的账号封禁，但需要注意的是如果封禁了第i个人，那么为了不打草惊蛇，就不能封禁他的关注者a_i。现在想知道最多可以封禁多少个xxs。 输入第一行是一个不超过300000的整数n，第二行是n个1到n的整数表示。 输出一行，一个整数表示答案。

```
01 #include<cstdio>
02 using namespace std;
03 #define MAXN 300005
04 int n, ans=0, a[MAXN], in[MAXN]={0};
05 bool vis[MAXN]={0};
06 void dfs(int cur, int w){
07     if(vis[cur])
08         return;
09     vis[cur]=true;
10     if(w==1) ans++;
11     __(1)__
12     if(__(2)__)
13         dfs(a[cur], __(3)__);
14 }
15 int main(){
16     scanf("%d",&n);
17     for(int i=1;i <=n; i++) {
18         scanf("%d", &a[i]);
19         in[a[i]]++;
20     }
21     for(int i=1;i <= n; i++)
22         if(!in[i]) __(4)__;
23     for(int i=1;i<=n; i++)
24         if(__(5)__) dfs(i,0);
25     printf("%d\n",ans);
26     return 0;
27 }
```

34. (3分) (1)处应填

A. a[cur]=cur; B. in[a[cur]]=0; C. in[a[cur]]--; D. in[cur]--;

35. (3分) (2)处应填

A. in[a[cur]] != 0 || w == 1　B. in[a[cur]]==0||w==0
C. in[a[cur]]!=0||w==0　D. in[a[cur]]==0||w==1

36. (3分) (3)处应填

A. 0　B. 1　C. w　D. 1-w

37. (3分) (4)处应填

A. dfs(i,1)　B. dfs(i，0)　C. dfs(a[i], 1)　D. dfs(a[i], 0)

38. (3分) (5)处应填

A. !in[i]　B. in[i]　C. !vis[i]　D. vis[I]

(2)

(烧作业)某课作业布置了N(3≤N≤10000)个题目，第i题对应的得分是a。作业的总得分的计算方式为去掉作业中得分最小的一个题，剩下其它所有题目得分的平均值。但很不幸小A遇到了一场火灾，前K(1<K<N-2)个题目被烧了，无法记录得分。小A想知道，K是多少时，可以得到最高的作业得分？作业被烧了前K页，这时的得分是从第K+1页到最后一页中，去除最小得分后取平均值。 输入第一行是整数N，第二行是n个不超过10000的非负整数表示a 输出一行，若干个整数表示答案。如果有多个K，请依次升序输出。

```
01 #include<cstdio>
02 #include<cmath>
03 #define min(a,b)(a<b?a:b)
04 #define MAXN 100002
05 using namespace std;
06 int n, k[MAXN], cnt=0;
07 int s[MAXN], minScore, sum;
08 double maxAverage=0,nowAverage;
09 int main() {
10     scanf("%d",&n);
11     for(int i=1;i<= n;i++)
12         scanf("%d",&s[i]);
13     minScore=s[n];
14     __(1)__;
15     for(int i=n-1;i>=2;i--) {
16         minScore=min(minScore,s[i]);
17             __(2)__;
18         nowAverage=__(3)__;
19         if(nowAverage> maxAverage){
```

```
            __(4)__
                maxAverage=nowAverage;
        } else if(fabs(nowAverage-maxAverage)<1e-6)
            __(5)__;
    }
    for(int i=cnt; i >= 1; i--)
        printf("%d\n",k[i]);
    return 0;
}
```

39. (3分) (1)处应填

A. sum=n　　B. sum=s[1]　　C. sum=s[n]　　D. sum=0

40. (3分) (2)处应填

A. sum=maxAverage*(n-i)　　B. sum+=s[i]
C. sum+=s[n-i]　　D. sum=s[i]+minScore

41. (3分) (3)处应填

A. (double)(sum+minScore)/(n-i)　　B. sum*1.0/(n-i)
C. (int)(sum-minScore)/(n-i)　　D. (double)(sum-minScore)/(n-i)

42. (3分) (4)处应填

A. k[++cnt]=i;　　B. k[cnt++]=i-1
C. cnt=1;k[cnt]=i-1;　　D. cnt=0;k[cnt]=I;

43. (3分) (5)处应填

A. k[cnt++]=i;　　B. k[++cnt]=i-1;　　C. k[cnt++] =n-i;　　D. k[cnt]=i;

CSP-J 初赛模拟卷（第 18 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

1．已知十进制中的18在x的进制中的表示为24，即(18)_10=(24)_x，则这个x进制为（ ）进制。

A．9　　B．8　　C．7　　D．6

2．根据网址的域名 http://www.jiangsu.gov.cn/，可以判断该网站是（ ）类型的网站。

A．商业机构　　B．军事机构　　C．政府机构　　D．组织机构

3．与计算机硬件关系最密切的软件是（ ）。

A．编译程序　B．操作系统　C．游戏程序　D．数据库管理程序

4．下列程序段执行后s的值为（ ）。

```
int i = 1, s = 0;
while (i++)
    if (!(i % 3)) break;
    else s += i;
```

A．6　　B．3　　C．2　　D．以上均不是

5．将19分解成3个不重复数字(1~9)之和(不计顺序)的方法有（ ）种。

A．6　　B．5　　C．4　　D．3

6．甲、乙、丙三位同学选修课程，在四门课程中，甲选修两门，乙、丙各选修三门，则不同选修方案共有（ ）种。

A．36　　B．48　　C．64　　D．96

7．已知某二叉树的先序遍历序列是 ABDCE，中序遍历序列是 BDAEC，则该二叉树的后序遍历为（ ）。

A．BDECA　　B．DBECA　　C．DBCEA　　D．BDCEA

8．计算机启动时，可以通过存储在（ ）中的引导程序引导操作系统。

A．ROM　　B．RAM　　C．Cache　　D．CPU

9．表达式a+b*c-(d+e)的前缀形式是（　）

A．abc*+de+-　　B．-+*abc+de　　C．-+a*bc+de　　D．abcde*++-

10．小军在家玩关灯游戏，小军家的灯有三种颜色分别是白、黄、红。按1下白灯亮，按2下灯灭，按3下黄灯亮，按4下灯灭，按5下红灯亮，按6下灯灭，再按又是白灯亮，以此循环。当按到49次和100次时灯的状态是（　）。

A．灯灭，灯灭　B．红灯亮，灯灭　C．白灯亮，红灯亮　D．白灯亮，灯灭

11．704与2048的最小公倍数是（　）。

A．45 056　　B．22 528　　C．180 224　　D．90 112

12．在（　）的情况下，逻辑 $A \vee B$ 运算的结果是逻辑“0”。

A．A和B全部是1　　B．A和B任一是1　　C．A和B全部是0　　D．A和B任一是0

13．小明夫妇请了小刚夫妇和小伟夫妇来他们家玩扑克。这种扑克游戏有一种规则：夫妇两人不能一组。小明和小红一组，小刚的队友是小伟的妻子，琳达的丈夫和小丽一组。那么这三对夫妇分别为（　）

A．小明-小丽，小刚-琳达，小伟一小红　B．小明-琳达，小刚-小红，小伟-小丽
C．小明-小丽，小刚-小红，小伟一琳达　D．小明-小红，小刚-小丽，小伟-琳达

14．4人过桥，每人单独过桥分别需要用时1分、2分、5分、10分，过桥需要灯(只有一盏)，一次只能2人一起过(意味着需要有人送灯回来)，过桥时间以用时多的人为准，则4人全部过桥时间最少需要（　）分。

A．15　　B．16　　C．17　　D．18

15．2000 年,华人学者姚期智因在计算理论(包括伪随机数生成、密码学与通信复杂度)方面的突出成就而荣获（　）

A．图灵奖　　B．奥斯卡奖　　C．诺贝尔奖　　D．普利策奖

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 **√**，错误填 **×**；共 **18** 小题，共计 **37.5** 分，各小题分值见题目）

(1)

```
01 #include<iostream>
02 using namespace std;
03 int main()
04 {
05     const int SIZE=10;
06     int height[SIZE],num[SIZE],n,ans;
07     cin>>n;
08     for(int i=0;i<n;i++){
09         cin>>height[i];
10         num[i]=1;
11         for(int j=0;j<i;j++){
12             if((height[j]<height[i])&&(num[j]>=num[i]))
13                 num[i]=num[j]+1;
14         }
15     }
16     ans=0;
17     for(int i=0;i<n;i++){
18         if(num[i]>ans) ans=num[i];
19     }
20     cout<<ans<<endl;
21     return 0;
22 }
```

判断题

16. (1.5分) 如果height数组中的输入有负数，则程序会出错。（ ）

17. (1.5分) 程序输出的 ans 小于或等于n。 （ ）

18. (1.5分) 将12行num[j]>=num[i]”改为“num[j]>num[i]”，程序的输出结果不会改变。（ ）

19. (1.5分) 将18行“num[i]>ans”改为“num[i]>=ans”，程序的输出结果不会改变。（ ）

单选题

20. (3分) 若输入的数据为 10 1 1 1 1 1 1 1 1 1 则程序的输出结果是（ ）。

A. 4　　B. 3　　C. 2　　D. 1

21. (3分) 若输人的数据为 10 3 2 5 11 12 7 4 10 15 6 因程序的输出结果是（ ）。

A. 4　　B. 5　　C. 6　　D. 7

(2)

```
#include<iostream>
using namespace std;
int n,m,i,j,p,k;
int a[100],b[100];
int main()
{
    cin>>n>>m;
    a[0]=n;i=0;p=0;k=1;
    do{
        for(j=0;j<i;j++)
            if(a[i]==a[j])
            {
                p=1;k=j;break;
            }
        if(p) break;
        b[i]=a[i]/m;
        a[i+1]=a[i]%m*10;
        i++;
    }while(a[i]!=0);
    cout<<b[0]<<".";
    for(j=1;j<k;j++) cout<<b[j];
    if(p) cout<<"(";
    for(j=k;j<i;j++) cout<<b[j];
    if(p) cout<<") ";
    cout<<endl;
    return 0;
}
```

判断题

22. (1.5分) 程序输入的n和m不能相等。（ ）

23. (1.5分) 程序输人的m不能等于0。（ ）

24. (1.5分) 第9~19行的do…while循环一共有2个出口。（ ）

25. (1.5分) 数组a和b中的数值都小于或等于n。（ ）

单选题

26. (3分) 若输入数据为“11 8”,则输出结果为（ ）

A. 0.375　　B. 0.(375)　　C. 1.375　　D. 1.(375)

27. (3分) 若输人数据为“5 13”，则输出结果为（ ）。

A. 0.386514　　B. 0.(386514)　　C. 0.384615　　D. 0.(384615)

(3)

```
01 #include <iostream>
02 using namespace std;
03 const int V = 101;
04 int n, m, ans, e[V][V];
05 bool visited[V];
06 void dfs(int x, int len)
07 {
08     int i;
09     visited[x] = true;
10     if (len > ans) ans = len;
11     for (i = 1; i <= n; i++)
12         if ((!visited[i]) && (e[x][i] != -1))
13             dfs(i, len + e[x][i]);
14     visited[x] = false;
15 }
16 int main()
17 {
18     int i, j, a, b, c;
19     cin >> n >> m;
20     for (i = 1; i <= n; i++)
21         for (j = 1; j <= n; j++)
22             e[i][j] = -1;
23     for (i = 1; i <= m; i++) {
24         cin >> a >> b >> c;
25         e[a][b] = c;
26         e[b][a] = c;
27     }
```

```
28      for (i = 1; i <= n; i++)
29          visited[i] = false;
30      ans = 0;
31      for (i = 1; i <= n; i++)
32          dfs(i, 0);
33      cout << ans << endl;
34      return 0;
35  }
```

输入表示一个没有重边、自环的无向图，顶点编号为 1 到 n，且 $1 \le n \le 100$、边权为正整数。

判断题

28. (1.5分) 第 19 行的输入中，如果满足 $m = n(n-1)/2$，则第 20～22 行的初始化可以省略。（ ）

29. (1.5分) 将第 31 行改为 `for (i = n; i >= 1; i--)`，程序结果不受影响。（ ）

单选题

30. (1.5分) 若输入如下，则程序输出为（ ）。

```
4 2
1 2 1
3 4 1
```

A. 8　　B. 4　　C. 2　　D. 1

31. (3分) 若输入如下，则程序输出为（ ）。

```
4 6
1 2 1
2 3 1
3 4 1
4 1 1
1 3 1
2 4 1
```

A. 1　　B. 3　　C. 5　　D. 7

32. (3分) 若输入如下，则程序输出为（　）。

```
4 3
1 2 10
2 3 20
3 1 30
```

A. 30　　B. 50　　C. 90　　D. 120

33. (3分) 若输入如下，则程序输出为（　）。

```
4 6
1 2 10
2 3 20
3 4 30
4 1 40
1 3 50
2 4 60
```

A. 60　　B. 100　　C. 150　　D. 200

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **10** 小题，共计 **30** 分，各小题分值见题目）

(1)

(高精度计算)由于计算机运算的数据范围表示有一定限制，如整型int表达范围是(-2^{31}~$2^{}$31-1)，unsigned long(无符号整数)是(0~2^32-1)，都约为几十亿，因此在计算位数超过十几位的数时，不能采用现有类型，只能自己编程计算。 高精度计算通用方法：高精度计算时一般用一个数组存储一个数，数组的一个元素对应于数的一位，将数由低位到高位依次存储在数组下标对应的由低到高的位置上。另外，申请数组大小时，一般考虑了最大的情况，在很多情况下表示有富裕，即高位有很多0，可能造成无效的运算和判断，因此一般利用一个整型数据存储该数的位数。下面的程序是一个高精度整数的加法运算，请补充完整程序。

```
01 #include <iostream>
02 #include <cstring>
03 using namespace std;
04
05 struct HugeInt{
06     int len;
07     int num[100002];
```

```
};
HugeInt a, b, w;
char c[100001],d[100001];
void Scan_HugeInt(){
    cin>>c;
    cin>>d;
    a.len=strlen(c);
    b.len=strlen(d);
    for(int i=0; i<a.len; i++)
        __(1)__;
    for(int i=0; i<b.len; i++)
        __(2)__;
}
void Plus(){
    w.len=max(a.len,b.len);
    for(int i=1; i<=w.len; i++) {
w.num[i] +=__(3)__;
w.num[i+1] +=__(4)__;
w.num[i]%=10;
}
if(__(5)__)
    w.len ++;
}

int main() {
    Scan_HugeInt();
    Plus();
    for(int i=w.len; i>=1; i--)
        cout <<w.num[i];
    cout <<endl;
    return 0;
}
```

34. (3分) (1)处应填

A. a.num[i]=c[i]-'0'　　B. a.num[a.len-i]=c[i]-'0'

C. a.num[a.len-i]=c[i]　　D. a.num[i]=c[i]

35. (3分) (2)处应填

A. b.num[i]=d[i]　　B. b.num[b.len-i]=d[i]

C. b.num[b.len-i]=d[i]-'0'　　D. b.num[i]=d[i]-'0'

36. (3分) (3)处应填

A. (a.num[i]+b.num[i])　　B. (a.num[i]+b.num[i])%10
C. (a.num[i]%10+b.num[i]%10)　　D. (a.num[i]+b.num[i]-10)

37. (3分) (4)处应填

A. w.num[i]/10　　B. w.num[i]　　C. w.num[i]%10　　D. w.num[i]-10

38. (3分) (5)处应填

A. w.num[w.len+1]>=0　　B. w.num[w.len+1]==0
C. w.num[w.len+1]!=0　　D. w.num[w.len +1]>1

(2)

(马走日)回溯算法实际上一个类似枚举的搜索尝试过程，主要是在搜索尝试过程中寻找问题的解，当发现已不满足求解条件时，就“回溯”返回，尝试其他路径。回溯法是一种选优搜索法按选向前索以达到目标。但当搜索到某一步时，若发现原先选择并优或达不到目标,则退回一步重新选择，这种走不通就退回去再走的技术称为回溯法，而满足回溯条件的某个状态的点称为回溯点。 马在中国象棋中以日字形规则移动。请编写一段程序，给定rxc大小的棋盘以及马的初始位置(m,n)，要求不能重复盘上的同一个点,计算马有多少途径可以遍历盘上的所有点。

```
01 #include<iostream>
02 using namespace std;
03 int r,c;
04 int cnt,tot;
05 int wayr[8]={2,2,1,-1,-2,-2,1,-1};
06 int wayc[8]={1,-1,2,2,1,-1,-2,-2};
07 bool mark[1001][1001];
08 bool check(int x, int y)
09 {
10     if(__(1)__) return true;
11     return false;
12 }
13 void search(int x,int y)
14 {
15     for(int i=0;i<8;i++)
16         if(__(2)__ && __(3)__)
17         {
18             mark[x+wayr[i]][y+wayc[i]]=true;
19             tot++;
```

```
            if(__(4)__) cnt++;
            search(__(5)__);
            tot--;
            mark[x+wayr[i]][y+wayc[i]]=false;
        }
}
int main()
{
    cnt=0;
    int m,n;
    cin>>r>>c>>m>>n;
    if(!check(m,n)) cout<<0<<endl;
    else if(r==1&&c==1) cout<<1<<endl;
    else
    {
        mark[m][n]=true;
        search(m,n);
        cout<<cnt<<endl;
    }
}
```

39. (3分) (1)处应填

A. x>=0 || y>=0 || x<r || y<c . B. x>=0 || y>=0 && x<r || y<e.
C. x>=0 && y>=0 && x<r && y<c D. x>=0&&y>=0|| x<r &&y<c

40. (3分) (2)处应填

A. check(wayr[i]，wayc[i]) B. check(x+wayr[i], y+wayc[i])
C. !check(x+wayr[i],y+wayc[i]) D. !check(wayr[I],wayc[i])

41. (3分) (3)处应填

A. mark[wayr[i]][wayc[i]] B. mark[x+wayr[i]][y+wayc[i]]
C. !mark[wayr[i]][wayc[i]] D. !mark[x+wayr[i]][y+wayc[i]]

42. (3分) (4)处应填

A. tot==r * c-1 B. tot-=r * c C. cnt==r * c-1 D. cnt==r * c

43. (3分) (5)处应填

A. wayr[i], wayc[i] B. x-wayr[i], y-wayc[i]
C. x-wayr[i], y+wayc[i] D. x+wayr[i], y+wayc[i]